

Capstone Project Report

(Project Term January – April, 2024)

Contextual Information Retrieval Assistant (CIRA) Powered by Local Neural Language Model (LNLM)

Submitted by

Ashish Subedi	Registration Number: 12018751
Sumit Paudel	Registration Number: 12018747
Sameer Khanal	Registration Number: 12002504
Gouransh Purohit	Registration Number: 12002138
Sanjeev Patel	Registration Number: 12001911
Aashish Bhandari	Registration Number: 12000405

Project Group Number: **CSERGC0241**

Course Code: **CSE445**

Under the Guidance of

Ashim Sharma

School of Computer Science and Engineering



DECLARATION

We hereby declare that the project work entitled **Contextual Information Retrieval Assistant (CIRA) Powered by Local Neural Language Model (LNLM)** is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering (Hons.) from Lovely Professional University, Phagwara, under the guidance of Ashim Sharma, during January to April 2024. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project group number: **CSERGC0241**

Name of Student 1: Ashish Subedi

Registration Number: 12018751

Name of Student 2: Sumit Paudel

Registration Number: 12018747

Name of Student 3: Sameer Khanal

Registration Number: 12002504

Name of Student 4: Gouransh Purohit

Registration Number: 12002138

Name of Student 5: Sanjeev Patel

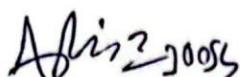
Registration Number: 12001911

Name of Student 6: Aashish Bhandari

Registration Number: 12000405

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B.Tech degree in Computer Science and Engineering (Hons.) from Lovely Professional University, Phagwara.



Ashim Sharma

Assistant Professor

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date : 2024/05/03



TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering (SCSE)

Program : P132-H::B.Tech. (Computer Science & Engineering) (Hons.)

COURSE CODE : CSE445

REGULAR/BACKLOG : Regular

GROUP NUMBER : CSERGC0241

Supervisor Name : Ashim Sharma

UID : 30054

Designation : Assistant Professor

Qualification : M.Tech (CSE)

Research Experience :

2 yrs

SR.NO.	NAME OF STUDENT	Prov. Regd. No.	BATCH	SECTION	CONTACT NUMBER
1	Ashish Subedi	12018751	2020	K20RE	123 6536536509
2	Sumit Paudel	12018747	2020	K20NY	6307557995
3	Sameer Khanal	12002504	2020	K20RE	9804145509
4	Gouransh Purohit	12002138	2020	K20DP	9079391341
5	Sanjeev Kumar	12001911	2020	K20NY	9956466658
6	Aashish Bhandari	12000405	2020	K20RE	8427682484

SPECIALIZATION AREA : Programming-I

Supervisor Signature:

PROPOSED TOPIC : Contextual Information Retrieval Assistant (CIRA) powered by Local Neural Language Model (LNLM)

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	7.00
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.00
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.04
5	Social Applicability: Project work intends to solve a practical problem.	7.04
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.41

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Dr. Vijay Kumar Garg	UID: 14085	Recommended (Y/N): Yes
PAC Member (Allied) Name: Ravi Kant Sahu	UID: 16920	Recommended (Y/N): Yes
PAC Member 3 Name: Salil Batra	UID: 16836	Recommended (Y/N): Yes

Final Topic Approved by PAC: Contextual Information Retrieval Assistant (CIRA) powered by Local Neural Language Model (LNLM)

Overall Remarks: Approved

PAC CHAIRPERSON Name: 25708::Dr.Rachit Garg

Approval Date: 12 Apr 2024

5/1/2024 9:31:31 AM

Table of Contents

1. Abstract	7
2. Introduction.....	8
3. Literature Review.....	10
4. Scope of the Project	13
5. Objectives of the project	14
6. Research and Methodology.....	15
8. Software Requirement Analysis.....	16
7. Functionality	17
8. Implementation	20
A. Interface	20
B. How website owners can create a new chatbot?	22
B.1. Signup Form.....	22
B.3. Login Form.....	23
B.3. Dashboard.....	24
B.4. Chatbot Creation	24
B.5. Chatbot information / Customization.....	25
B.6. Chatbot Statistics.....	26
C. Working of the Chatbot	26
9. Source Code	29
9.1. Login Page	30
9.2. Signup Page	31
9.3. Dashboard Page	32
9.3. Create Chatbot	33
9.4. Manage Chatbot	34
9.5. Chatbot Design.....	35
9.6. Chatbot Logic.....	36
10. Results and Observations.....	37
11. CIRA Deployment.....	38
11. Conclusion	39
11. Publication Details	40

12. Award	41
13. References.....	42
14. Plagiarism Report.....	43

1. ABSTRACT

Companies are increasingly using chatbots to provide customer service but the current generation of chatbots lack the ability to resolve the issue of the query without redirecting to the human agent. The most they can do is answer the FAQs which are already present in the website. Through, Contextual Information Retrieval Assistant (CIRA), an AI powered chatbot tailored to specific websites. Our approach integrates website content scraping, natural language processing techniques powered by open-source LLMs, and customization methods. This allows rapid, cost effective deployment of intelligent chatbots capable of providing relevant and informative user support at a higher level than the traditional chatbots. Our platform can adapt to a diverse set of domains, including educational institutions, healthcare providers, hosting platform providers as well as many other service and product related organizations, optimizing and enhancing overall user experience. CIRA permits users to extract answers directly from the websites, get rid of the need for time consuming searches, and page hopping while being not able to find as simple a thing as an international student per credit fee for a graduate student in a Data Science program. Why this specific problem? Because out of the 20 US aspiring students surveyed in Lovely Professional University, 2 students could find this exact information in 3 minutes and 14 in under 5 minutes, and for others, it took a remaining whopping 20 mins, when we compared that with CIRA it was possible in only a few seconds based on your typing speed.

Leveraging domain specific data and amalgamating it with powerful AI resources available in the tech world right now, CIRA provides accurate and relevant information tailored to individual needs while also having security and privacy as its USP, on top of that still having a lot of potential for future growth, we believe CIRA the future of chatbots.

2. INTRODUCTION

The 21st century has seen a quantum leap in technology. The advancement of technology in just 5 years dwarfs that of the last century. According to Moore's law, the number of transistors in a circuit doubles every two years, increasing the computational speed by manyfold. This increase in power has led to the creation of large data sets that can be used to create and train AI.

The recent progression in Artificial Intelligence has borne a fruit that humans as a whole can benefit from. Thanks to Artificial Intelligence, humans can now imagine the possibilities that were labeled as just theories in the past. A single AI bot now has the ability to outdo an entire team of developers. According to World Economic Forum around 40% of working hours across industries can be impacted by LLMs.

LLMs which stands for large language models is an AI algorithm that utilizes a vast amount of data to understand and summarize content. These Large Language Models programs can then generate new content and ideas based on the data that is studied. As the amount of data increases the productivity and precision of LLMs get sharper.

Modern LLMs use neural networks or transformers. Given a large number of parameters and the transformer model, these models can understand and generate accurate responses rapidly, making AI technology broadly applicable across many different domains. In the paper published by the World Economic Forum in 2021, sectors such as banking, insurance, and engineering show a high potential for automation through AI.

Banking sectors alone can automate 50% of their work, improving the productivity rate with a fraction of the cost. The World Economic Forum also projects that over 2.3 million total employees will be replaced by AI in China alone.

The creation of Chat-GPT marks a historic moment in the AI shift. Chat-GPT was created by OpenAI in November of 2022. Chat Generative Pre-training Transformer uses a neural network to understand the prompt given by the user and will attempt to answer it in the context of the data it was trained on. According to an experiment conducted by the Massachusetts Institute of Technology, out of 453 college students who were given a task to write an article, those who used Chat-GPT completed the task 11 minutes faster than those who didn't use ChatGPT. The quality

of the article also increased by 18 percent. This shows the potential of Artificial Intelligence in human productivity.

Artificial Intelligence-integrated chatbots have started to emerge as a new way of perceiving website content. Such chatbot has eliminated the need to skim through the entire website saving valuable time and effort. Getting answers that are easy to understand and assistance in finding online self-service resources, instead of having to wait in line or wait for customer service to open, were reported as key attractors to using chatbots for customer service. The preference for websites with AI-integrated chatbots is on a steady rise. Even though chatbots have been in use for decades, the need for a human to generate a response made it less responsive. However, with the advancement of AI, this problem can be tackled as the responses are auto-generated by the AI with the help of data it was trained on. Contextual Information Retrieval Assistant (CIRA) has been designed in a similar way to gather information about a website and answer the queries given by the user, saving time and effort for the users.

Contextual Information Retrieval Assistant (CIRA) was created to solve the issues we have with the current chatbot systems, they lack something in particular which is decision-making. This used to be a fairly complex topic before the concept of LLMs but now, as we have access to different kinds of open source as well as paid models provided by different companies, we can utilize this particular concept to improve the current chatbot experience.

CIRA under the hood utilizes the LLaMA2 model provided by Meta to create an intelligent conversational agent that is capable of handling diverse inquiries effectively. Users can effectively train their own data by providing their website, which CIRA is able to crawl and extract information. This research project seeks to explore the potential of AI in enhancing user engagement and experience in the online world.

3. LITERATURE REVIEW

In one research, it was found that College students often struggle to find information across campus departments. GALGOBOT, a website chatbot, aims to solve this by offering easy access to updated information through login, natural language processing, and a user-friendly interface. It benefits all students, especially international applicants, by allowing them to inquire about fees and academics without visiting offices. Future plans include expanding information, improving login, and even offering online exams through the chatbot. The basis of this research paper and the project is inspired by this development. GALGOBOT is limited with its development in just the school sector whereas, CIRA is applicable everywhere and adds additional information, if asked, with the help of its LLM.

How do you want your Chatbot? Another research study explored by Indrani Medhi Thies, Nandita Menon, Sneha Magapu, Manisha Subramony, and Jacki O'Neill titled "How Do You Want Your Chatbot? An Exploratory Wizard-of-Oz Study with Young, Urban Indians" states how young Indians prefer chatbots to act (using Wizard-of-Oz studies). Users liked a helpful and friendly bot ("Maya") but also wanted playful or supportive traits. Ideally, a chatbot would adapt its personality to become a trusted friend users can talk to anytime, anywhere. This research highlights the need for user preferences in chatbot design. In this regard, a personality that adapts to the user could be ideal.

In another study titled "Jollity Chatbot- A contextual AI Assistant" by Kanakamedala Deepika, Veeranki Tilekya, Jatroth Mamatha, Subetha T explores on how it has been found that the purpose of Chatbots in recent times is not just to get things done; some are even designed to be our friendly companions, offering emotional support and entertainment. The Jollity Chatbot, built with Rasa, personalizes interactions and shows promise in understanding users' needs. Future plans include incorporating real-life examples and voice chat, aiming to provide a free, supportive, and uplifting environment for users going through rough times. Furthermore, research suggests that chatbots can be effective in mitigating loneliness and social isolation, particularly for those who may have difficulty connecting with others.

In another study by Eleni Adamopoulou and Lefteris Moussiades titled “An Overview of Chatbot Technology” they observed that the rapid evolution of chatbot technology across various domains, from marketing to healthcare, has garnered significant interest globally. This paper provides a comprehensive overview of this evolution, tracing the historical trajectory of chatbot adoption and discussing the diverse motivations driving their usage. Highlighting chatbots' utility in different sectors, the paper underscores their potential to minimize human intervention in device usage and enhance accessibility, particularly through messaging apps. Moreover, it delves into the influence of social stereotypes on chatbot design and elucidates essential technological concepts underlying their functionality. The classification of chatbots based on various criteria, alongside an exploration of their general architecture and prominent creation platforms, further enriches the understanding of their versatility and applicability. As technology advances, chatbots are poised to become more proficient information-gathering tools, potentially blurring the distinction between human and bot interactions. While offering substantial benefits such as operational cost savings in customer service, they also raise ethical considerations regarding abuse and deception. This research serves as a valuable resource for users and developers alike, equipping them with a deeper understanding of chatbots' principles and functionality. Future endeavors could delve into detailed comparisons of existing chatbot platforms and explore ethical dimensions, shedding light on issues like abuse and deception in human-bot interactions.

With the development of any new technology, there are many challenges arising. One of the research titled “Challenges in Chatbot Development: A Study of Stack Overflow Posts” by Ahmad Abdellatif, Diego Costa, Khaled Badran, Rabe Abdalkareem, Emad Shihab analyzed the Stack Overflow discussions to understand the difficulties encountered in the development of Chatbots. Topic modeling was used as a common theme to identify similar problems for developer inquiries. The most difficult hurdle was training the NLU model of the chatbot while creating and integrating the chatbots were the most widely discussed topics. This can guide future research to develop solutions for these challenges, enabling platform creators and instructors to more effectively assist chatbot developers. The study acknowledges its limits and intends to investigate discussions in other forums and analyze chatbot code to gain a more comprehensive picture. Another author suggests that, with the growing popularity of chatbots, its underlying architecture should be designed keeping the security and privacy as a top priority. Similarly, the author discusses the potential vulnerabilities that may arise within different sections of the Chabot architecture such as

vulnerabilities related to client module, communication module, response generation, and database module. Without proper testing of such susceptible chatbots, there could be chaos in the community. Hence, it is important to conduct a full analysis of the Chatbot's vulnerabilities before deployment.

Since CIRA is an advanced form of chatbot-powered LNLM to further add up the value it has been found from research conducted by the Faculty of Engineering and Informatics, Universitas Multimedia Nusantara, Indonesia showed that the integration of chatbot powered by large language model (OpenAI model) with Journal Business Management, and Social Studies (JBMS) website allowed seamless search of the desired journals faster than before which brought new level of experience to the users while scoring an average User Acceptance Testing (UAT) score of 4.14 out of 5, emphasizing higher user satisfaction.

4. SCOPE OF THE PROJECT

Chatbots today are pretty limited in what they can do. To fix that and make them give better answers, we've been using these open-source language models called "Large Language Representations" (LLMs for short). These LLMs are a big game-changer because they significantly improve how chatbots understand language and respond to people.

With LLMs, chatbots can finally grasp the complexities of human speech, leading to more accurate and helpful answers. They can also learn and adapt based on how people interact with them, getting better over time and making the whole experience smoother. This means chatbots can offer more personalized and informative support in various fields like education, healthcare, and even website hosting.

LLMs also allow chatbots to tackle specific tasks efficiently, like retrieving detailed information like international student fees. This creates a more comprehensive and specialized interaction for users.

By using these open-source LLMs, companies can ditch the limitations of old-school chatbots and provide a more sophisticated and interactive customer service experience. These advanced AI models empower chatbots to not only understand complex questions but also give answers that are relevant to the conversation, bridging the gap between human-like interactions and simple support.

This improved language processing ability allows chatbots to provide more accurate and detailed information, leading to higher user satisfaction and engagement across various sectors.

Basically, using LLMs streamlines how chatbots work, makes them more productive, and helps them deliver tailored answers that meet people's needs more effectively.

Overall, integrating open-source LLMs into chatbots is a major leap forward in chatbot technology. It solves the limitations of current chatbots and allows businesses to revolutionize their customer service by providing more relevant, informative, and efficient support. This approach not only improves the customer experience but also opens doors for chatbots to excel in specific tasks, showcasing the potential of AI to completely transform how we interact with businesses and access services across various industries.

5. OBJECTIVES OF THE PROJECT

The basis of our report is the development of a chatbot system which tailors' specific websites and can be integrated into them really easily. How this works is CIRA extracts all the relevant information from the added website and answers those questions. The basic question arises when we decide how we can make the chatbot output a correct response based on what user is really asking. To solve this issue we started using LLM models. Especially Llama from Meta, which is also open source and provided it's own intelligence.

Using this method, CIRA can now generate pretty good responses which are accurate and relevant to which user asked. Another USP of CIRA is that we are trying to make it more secure as possible. People in different businesses don't want their data to be shared with others and that's why we only train the data on our local machines. Instead of using APIs like Open AI, using a local LLM like Llama was our choice. This way we are not sending confidential data to the clouds of big corporations and our clients can stay satisfied.

Our research uses LLaMA2 model from meta to create CIRA. The intelligence of LLaMA2 powers our chatbot, it was able to filter out relevant information from the provided context and respond to user by mixing it's own knowledge and the knowledge it got from the provided context.

The main hurdle of the project was the way to extract the information from the websites. Asking clients data would be a pretty heavy task so we ended up using a nifty web scraping as all the required data are already present in the website and are public but clients can also give us special access to their data. This allowed CIRA to be personally customized for that specific website which the data was scraped from. The scraped data is also cached so, if the original source goes offline and to save bandwidth, the cached data is used. How this algorithm works is explained below in this report.

User engagement is a huge metric in web development. Every company wants visitors to stay long as possible on their website. We know, by reducing page hooping people can get information in seconds and keep user engaged improving user engagement in websites. AI chat and information is quite common these days, but only big companies can easily implement them. Our goal is that even small companies can enjoy the benefits of smart AI on their website. CIRA is a promising and innovative solution for the next generation of chatbot technology.

6. RESEARCH AND METHODOLOGY

We wanted to make sure that our research was thorough, so we ended up using a different variety of methods for CIRA.

We conducted a survey with some students in B.Tech CSE students from Lovely professional university. The whole results is mentioned below in this report. There were 25 students to be exact. We gave them a task, which was to find specific information online. The specific task was how much data science credit costs at the University of Texas at Arlington. The results for the task was pretty astonishing. A great chunk of the students used CIRA and were able to find the answer in just seconds whereas they took 10-20 minutes with page hopping. This highlights how effective AI powered chatbots are.

Our research wasn't just with the surveys. We dived deep into different research papers and analyzed case studies, by looking closely at security and privacy issues. We combined both quantitative and qualitative data by looking closely at the people already done in the past and establishing our own analysis. This way we found many different valuable insights and we were able to confirm what we were proceeding with CIRA. This way we were able to build a solid foundation for developing and implementation of our project.

We also incorporated a Text Generation Methodology. This term sounds fancy but it's really. It just means we used different special tools to create summaries, and responses to the data we were studying. It made data analysis and content creation seamless in the whole research project. This was the reason our research became more efficient and accurate, and we were able to present everything in organized and clear format.

The methods we used were all over the map. We used surveys, we analyzed different research projects, did many security checks, case studies and text generation. This approach helped us to explore everything from the user's perspective on their tech needs and capabilities. This approach helped us develop and refine CIRA into the current innovative and focused AI chatbot.

8. SOFTWARE REQUIREMENT ANALYSIS

The ability to scrape and extract relevant information from the provided website was the foremost functional requirement of CIRA. The scraping first included all the internal links of a website along with their anchor tags. Then when user asks any kind of query we would find the relevant link amongst the collected links and search them for information again by scraping. This time we extracted all the texts and stripped all the html tags so that we only get relevant information. We are able to gather necessary information for good responses for the chatbot this way.

Utilizing Natural Language Processing is our another functional requirement. NLP which are powered by open source Large Language Models like LLaMA are the foundation of this chatbot. This allowed CIRA to understand directly what user meant when they asked the question. The limitation in the traditional chatbot was instead of understanding the user they just sent what answer matched what question but with our method we were able to understand the user and give them a proper answer.

Another key feature we wanted to establish was the ability to customize our chatbot. Chatbot functional well is a key feature but as people who love good presentations we couldn't help ourselves from making it shine on what matters, the presentation. The main user engagement lies of familiarity. If the chatbot's style and the website it is embed on looks very different then people become hesitated on using it. Making both same gives them a sense of familiarity. This enabled us to ensure a branded user experience.

Our CIRA is able to adopt to different diverse set of domains like educational institutions, healthcare, hosting platforms and much more sectors you name it. The ability to adapt to those various sectors makes CIRA optimize and enhance overall user experience, we are able to cater the unique need and expectations of different user groups and stakeholders.

Our research paper focuses on security and privacy this it is our unique selling point. CIRA was built to emphasize the user data protection and ensure the provided data is accurate and relevant by maintaining the confidentiality of user's as well as the businesses' personal information.

7. FUNCTIONALITY

A. How CIRA chatbot can be created.

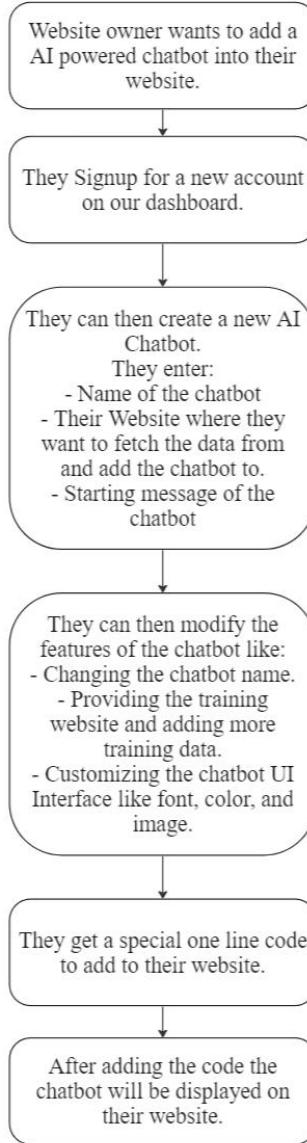


Figure 1: CIRA Prototype flow diagram on how the website owners can create their own chatbot system from CIRA Dashboard.

From the above Prototype Figure 1, it can be observed that website owners can easily add CIRA to their websites. The first step would be to sign up for an account. Then, they can create their chatbot. They need to provide URL of their website, while creating the chatbot. It can also be noticed that the creator has every control on the created chatbot, they can edit the colors, images, and fonts. After the chatbot is active, the owners get a special link to add it to their website, and the chatbot starts working.

B. How does CIRA chatbot work?

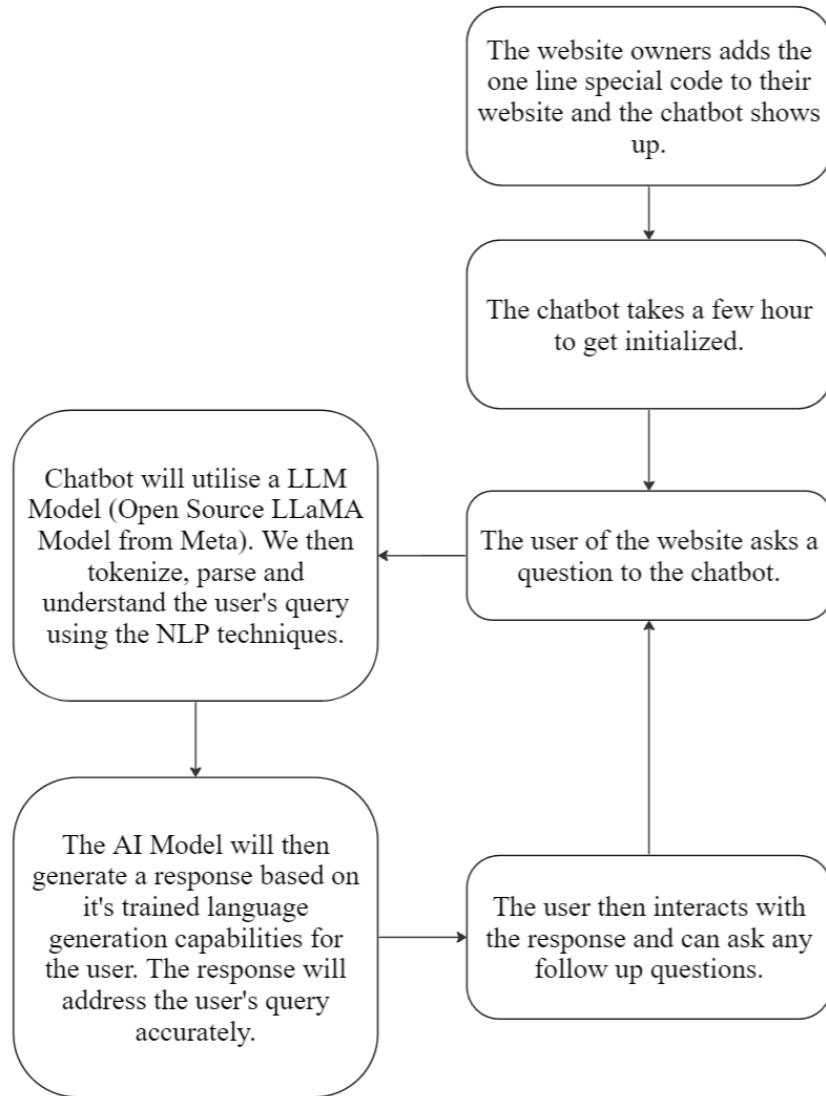


Figure 2: CIRA Working Prototype flow diagram on how CIRA Works behind the scenes when a website user asks a question.

As illustrated in Prototype Figure 2, The chatbot starts working when it is embedded to the website provided by the website owner. To embed the chatbot the website owner needs to copy the provided script and paste it to their website. It takes a few hours to get initialized. The initialization process is illustrated in prototype Figure 3. When a user asks the chatbot a query, the chatbot uses the data it obtained from scraping the website, then sends the data to the Open-Source LLaMA Model along the user's query. CIRA then provides the relevant answer based on the data. The user then can ask any follow up questions.

C. How is the data trained in CIRA?

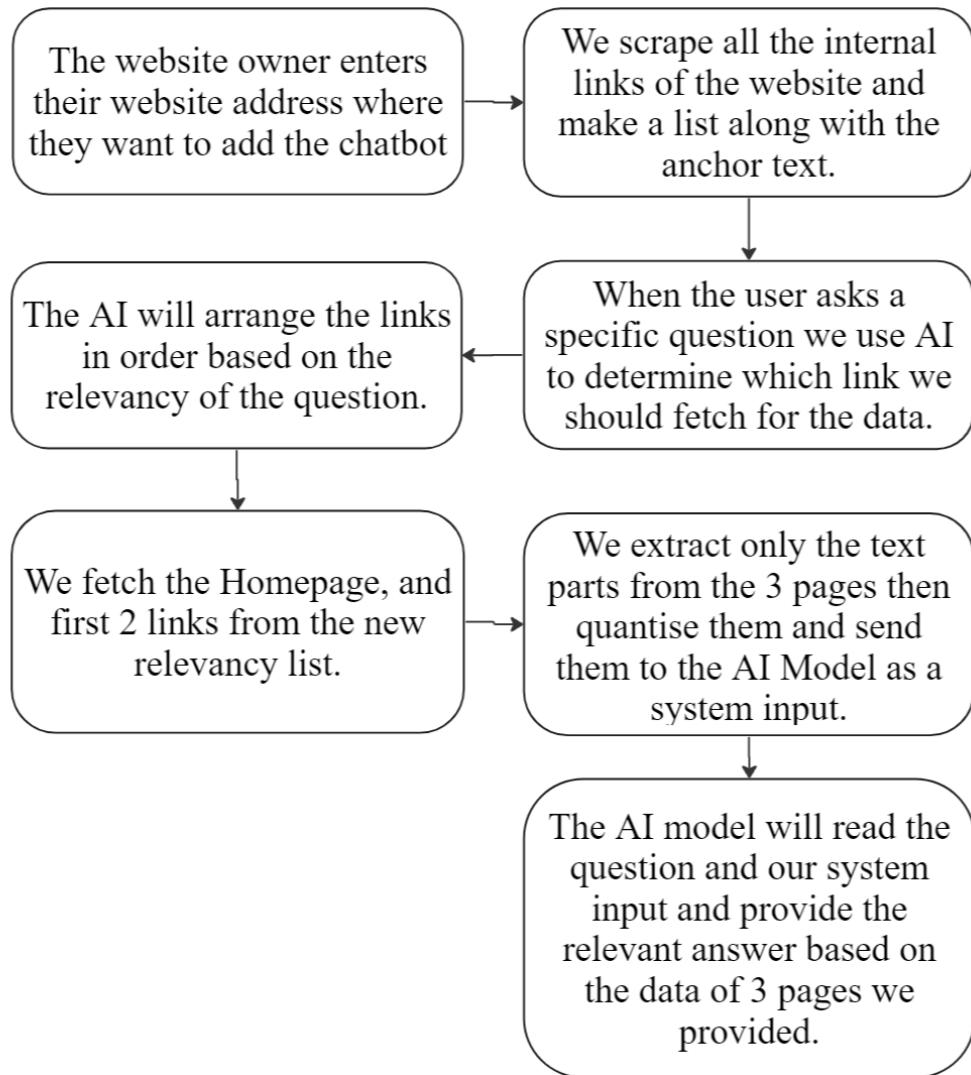


Figure 3: CIRA Training Prototype flow diagram on how CIRA works behind the scenes to train the data from the provided website.

From the above prototype Figure 3, it can be understood how the response for the chatbot is generated. When the creator enters their website URL in the CIRA dashboard, all the internal links are scraped by CIRA bot. Those links are stored in the database along with their anchor texts. When a user asks a question in the CIRA Chatbot, CIRA scans all the previously extracted URLs and their anchor texts and generates the most relevant links according to the asked question. The contents of the first 2 links and the homepage of the website are extracted and sent to the AI model as system input along with the question asked by the user as user input. CIRA would then read the question and based on the system input it will analyze and generate a relevant answer to that question based on the system input.

8. IMPLEMENTATION

A. INTERFACE

The Interface and Dashboard of CIRA is developed using HTML, CSS, JS, and PHP Laravel Framework. Laravel is a very good framework that is used by many developers worldwide to create fast and scalable web applications. It provides different built-in features like Eloquent ORM and migrations which makes the development process seamless.

Settings for Aeonfree CIRA Bot (AFB)

Copy Bot Code

```
<script src='http://127.0.0.1:8000/chatbot/1/embed/script.js'></script>
```

Update Bot Information

Bot Name:

Logo: No file chosen

Initial Message:

Website to be used for the bot:

Message placeholder:

Enable voice messages:

Header Color: Active Color:

Body Color: Inactive Color:

Text Color: Text Color:

Font Family:

Box Shadow Intensity:

Bot Example

Type a message here

Powered By **CIRA**

Figure 4: Dashboard Design of CIRA where the website owner can change colors, fonts, texts, etc. of the chatbot.

As seen on Figure 1, the intention behind CIRA is to be a general use case chatbot that can be easily used by website users and can be integrated with any website without any effort from the website owners. As illustrated in Figure 4, the implementation process is simplified as much as possible to make the chatbot highly customizable. The website owners should be able to modify every aspect of CIRA Chatbot from colors and fonts to even the intensity of the drop shadow of the chatbot on their website.

CIRA works by interacting with the LLaMA 2 AI model from Meta. It uses the local instance of the model hosted on different cloud servers which will run as API microservice. This way the data of the clients are not shared with any other 3rd party Companies like Open AI. How this process works is explained in the prototype flow chat design Figure 3.

The chatbot interface consists of 3 components, Header, Main, and Footer. The header part includes the image of the chatbot, name, and status of the chatbot. The main component showcases all the messages we sent to the chat and all the responses we got. We can see the timing of the messages and either like or dislike the response provided by the chatbot as well for user content level. The footer section includes an input form where a message can be written by the visitor and a send button. Below the input form is the personal branding of CIRA. Outside the chatbot, we have provided a simple toggle button to show the chatbot when clicks or closes it.

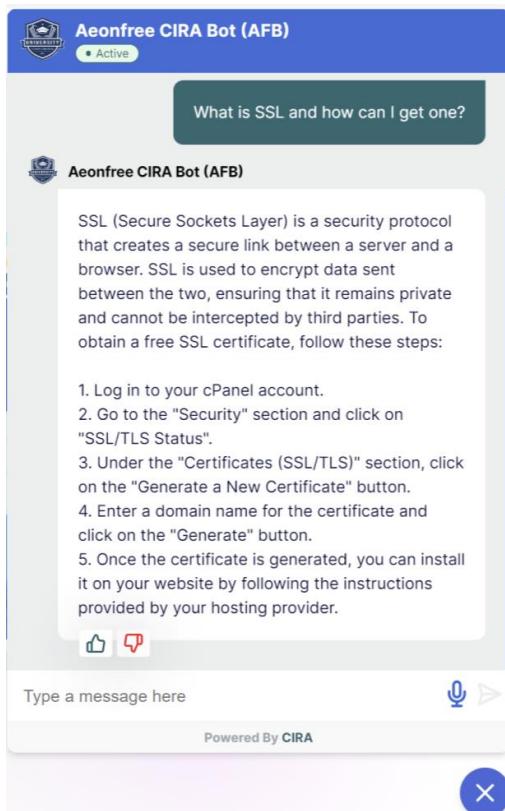


Figure 5: Chatbot Design of CIRA.

The complete design of the chatbot UI can be seen in Figure 5. It can be observed that a user asks a question about “SSL Certificate” and the chatbot provides an appropriate response for the query.

B. HOW WEBSITE OWNERS CAN CREATE A NEW CHATBOT?

It can be noticed from Prototype Figure 1, Creating a new chatbot based on CIRA is easy, there is a simple Signup form, where Individuals can easily sign up for an account.

Individuals can also log in through the login page if they already have an account. After creating an account or logging in, a new Chatbot can be easily created. Currently, individuals can create as many chatbots as they like, there is no limit to it.

B.1. SIGNUP FORM

The signup form is quite simple, the user needs to just enter his first name, last name, email address, and password. After creating an account they can signup to their account where they can create and manage CIRA.

The screenshot shows the 'Signup for free CIRA chatbot.' form. It contains fields for 'Your Name', 'Your email address', 'Password', and 'Confirm your password'. There is a checkbox for agreeing to 'Terms & Conditions' and a 'CREATE ACCOUNT' button. Below the form is a section for existing users with 'Already have an account?' and a 'LOGIN!' button. At the bottom is a footer with a gear icon.

Signup for free CIRA chatbot.

Your Name:

Your email address:

Password:

Confirm your password:

Do you agree to [Terms & Conditions](#)

CREATE ACCOUNT

Already have an account?

Login to enjoy all the dashboard features.

LOGIN!

B.3. LOGIN FORM

Website owners can Login to CIRA from the login form. After logging in they can create a new chatbot or manage already created ones.

Login to your dashboard

Your email address:

Password:

Remember me

LOG IN

[Forgot your password?](#)

Don't have an account?

Signup now to get free chatbot from CIRA.

SIGNUP NOW!



Copyright © 2024 | Made With ❤️ And Smile :)

B.3. DASHBOARD

After logging into the dashboard users can see all the chatbots they have created. They can then manage their account or manage their chatbots from the dashboard.

The screenshot shows the CIRA dashboard interface. On the left sidebar, there are links for 'Home', 'User Settings', and a 'Logout' button. At the top, there are navigation links for 'HOME', 'YOUR BOTS', and 'PROFILE'. On the right, there is a user profile section for 'Sameer Khanal' with an email address 'admin@rudrax.net' and a notification bell icon. The main content area displays a message: 'To Create a new Hosting account, please click on the (+ New Account) button.' Below this, a table titled 'Your Chat Bots' lists three entries:

Bot (4)	Status	Manage
Aeonfree CIRA Bot (AFB) aeonfree.com	ACTIVE	MANAGE
Hello Bot demo aeonfree.com	ACTIVE	MANAGE
Lovely Professional University Bot lpu.in	ACTIVE	MANAGE

At the bottom of the page, a copyright notice reads 'Copyright © 2024 CIRA | Created With ❤️'.

B.4. CHATBOT CREATION

If the user wants to create a new chatbot they can click on + New Account and create a new chatbot.

To create a new chatbot they need to add a Name to the chatbot and some description. They then need to add a url of their website.

The screenshot shows the 'Enter the bot details' form. On the left sidebar, there are links for 'Home', 'User Settings', and a 'Logout' button. At the top, there are navigation links for 'HOME', 'YOUR BOTS', and 'PROFILE'. On the right, there is a user profile section for 'Sameer Khanal' with an email address 'admin@rudrax.net' and a notification bell icon. The main content area has a heading 'Enter the bot details' and a sub-instruction 'Add your bot information'. It contains four input fields:

- Bot Name: CIRA for aeonfree
- Bot Description: Helpful bot for aeonfree.com
- Website to be used for the bot: aeonfree.com

At the bottom of the form is a blue 'CREATE BOT' button.

At the very bottom of the page, a copyright notice reads 'Copyright © 2024 CIRA | Created With ❤️'.

B.5. CHATBOT INFORMATION / CUSTOMIZATION

The created chatbot can then be customized to match the user's preferences. After customizing them they can embed the chatbot to their own website by copying and pasting the provided JavaScript code. They can also view their chatbot statistics by clicking on view statistics link.

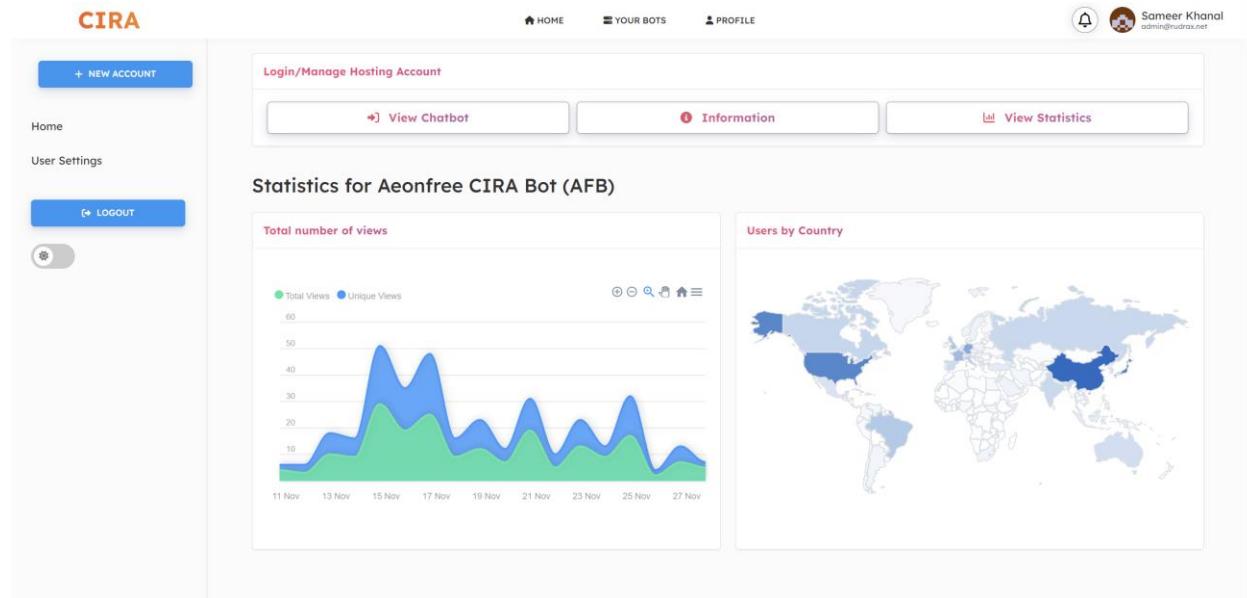
The screenshot shows the CIRA platform interface for managing chatbots. At the top, there are navigation links for HOME, YOUR BOTS, PROFILE, and a user profile for 'Sameer Khanal'. Below the navigation is a 'Login/Manage Hosting Account' section with buttons for 'View Chatbot', 'Information', and 'View Statistics'. The main content area is titled 'Settings for Aeonfree CIRA Bot (AFB)'. It includes sections for 'Copy Bot Code' (with a code snippet and a 'COPY' button), 'Update Bot Information' (with fields for 'Bot Name' set to 'Aeonfree CIRA Bot (AFB)', 'Logo' (choose file), and 'Initial Message' set to 'I am a Helpful bot for aeonfree.com'), and a 'Bot Example' preview window showing the bot's welcome message. The bottom half of the screen shows a detailed configuration panel with sections for 'Initial Message' (set to 'I am a Helpful bot for aeonfree.com'), 'Website to be used for the bot' (set to 'aeonfree.com'), 'Message placeholder' (set to 'Type a message here'), and various styling options like 'Enable voice messages', color swatches for Header, Body, and Text colors, font families (Lexend Deca), and font shadow intensities. A large blue 'UPDATE BOT' button is at the bottom of this panel. To the right of the configuration panel is a message input field with a placeholder 'Type a message here' and a send button.

While creating a chatbot the creator needs to specify its name, a description (or welcome message), and the website URL which is used to fetch the data for the chatbot to train and as well as show the chatbot. All the information can be changed later from the chatbot's Information page. After the chatbot is created, by going to the information page it can be customized further by changing

its text colors, background colors, fonts, logos, text messages, titles, welcome message, height, width, and where the owners like to show the chatbot and much more.

B.6. CHATBOT STATISTICS

After the chatbot is created and initialized the user can view the statistics from the View Statistics page. The users can see the total number of views in a pretty graph and the users of the chatbot according to the country.



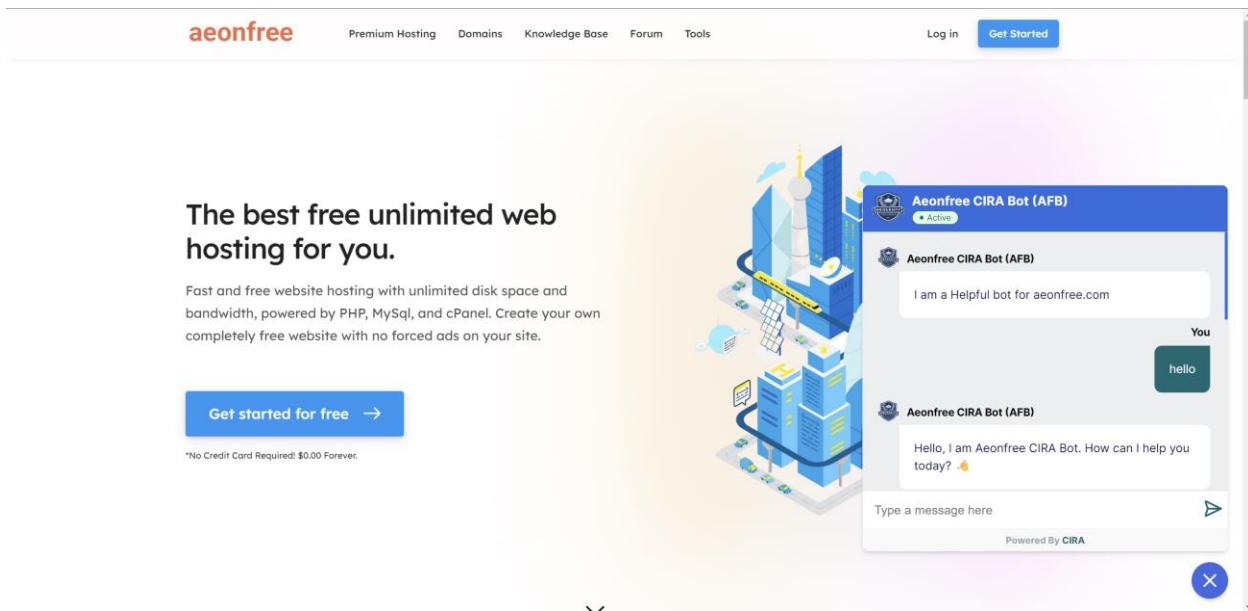
C.WORKING OF THE CHATBOT

After a chatbot is created by a website owner, it starts scraping data from the mentioned website. Prototype Figure 2 shows how this process works. The website owner can also specify additional links to the ones the bot has already scraped or additional information they want to feed the bot.

A link to embed the chatbot will be provided to the website owner. The website owner then can copy and paste the link to their website and the chatbot will start appearing on their website.

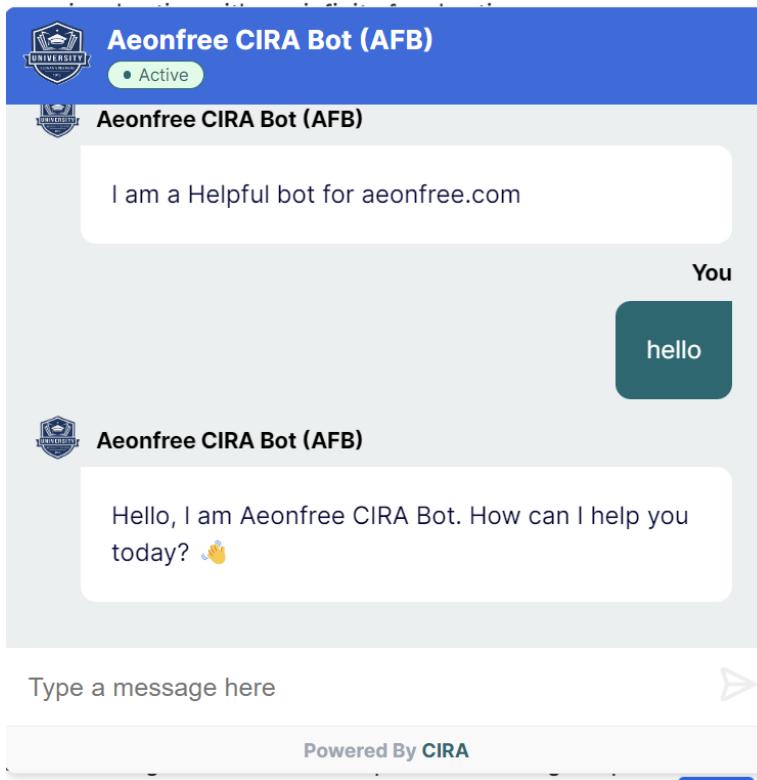
The website owners then can see the statistics of the chatbot through the dashboard provided by CIRA. The statistics include the countries of the users who are asking the queries to the chatbot, the messages being sent to the chatbot, and the user content level.

CIRA also provides a feature to assign a human behind the chatbot, but instead of sending responses directly to the chat user, it is sent to the human, and the human can choose an appropriate response based on multiple generated responses and send it to the user. It is helpful in cases where chatbots need some kind of human intervention.

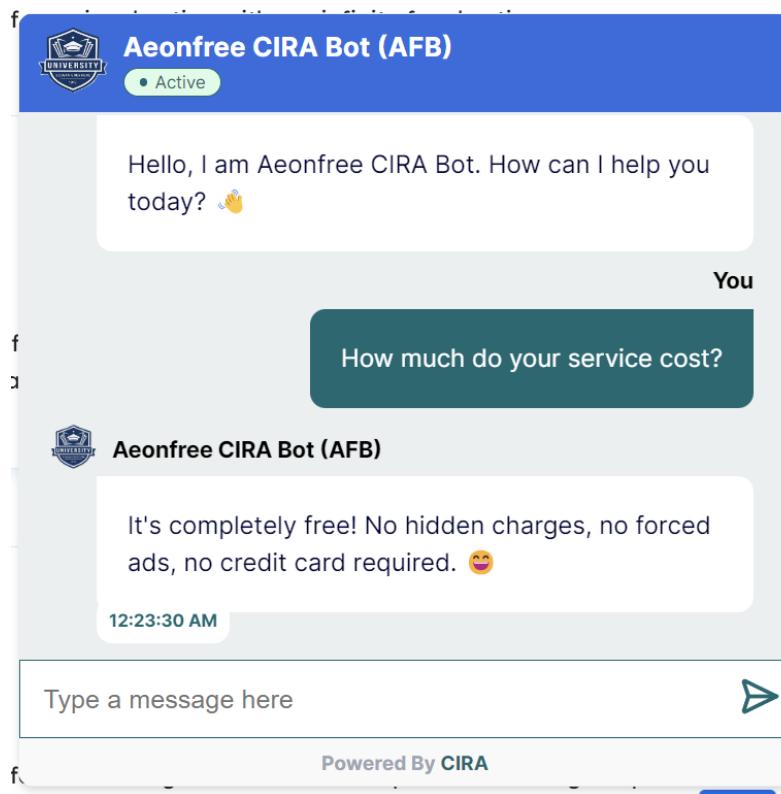


When the user copies and pastes the provided JavaScript page on their website they will see our chat popup widget immediately. It will look exactly like on the above figure. CIRA was installed on a free hosting service called Aeon Free for around 30 days and the above screenshot was taken from the website. Below are some of the questions, CIRA was asked.

CIRA can respond to simple greetings.



You can ask basic questions to CIRA and it will respond flawlessly without any issues and page hopping. This is the information which was on the homepage and CIRA responded it really quickly.



To make it more difficult we asked CIRA a question which required 4 page hoppings to find. Homepage > Blog > Tutorials > How to install SSL Certificate

CIRA was able to understand the question and provide a proper response based of the page it found through our algorithm.

It was able to find a correct answer, then generate a response and sent it as the response.

You can also like or dislike responses provided by CIRA. The likes and dislikes gets stored in the database and will be used to improve the responses provided by CIRA in the future.

The screenshot shows a mobile application interface for 'Aeonfree CIRA Bot (AFB)'. At the top, there's a blue header bar with the bot's logo on the left and the text 'Aeonfree CIRA Bot (AFB)' in white. Below the logo, a green button indicates the bot is 'Active'. The main content area has a dark teal background with white text. It asks 'how to instal a SSL certificate on my website?'. Below this, the bot's name 'Aeonfree CIRA Bot (AFB)' appears again with its logo. The main text is a guide: 'To install an SSL certificate on your website:' followed by a numbered list from 1 to 7. At the bottom, there's a text input field 'Type a message here' and a green send button with a white triangle icon.

how to instal a SSL certificate on my website?

Aeonfree CIRA Bot (AFB)

To install an SSL certificate on your website:

- 1 Go to your Dashboard.
- 2 In the Dashboard, click on "SSL Certificate."
- 3 Click on the "Create a New Certificate" button.
- 4 Enter your website URL.
- 5 Click on "Generate Certificate"
- 6 Click on the "Create" button.
- 7 On the next page, click on the "Install Certificate" button

Type a message here

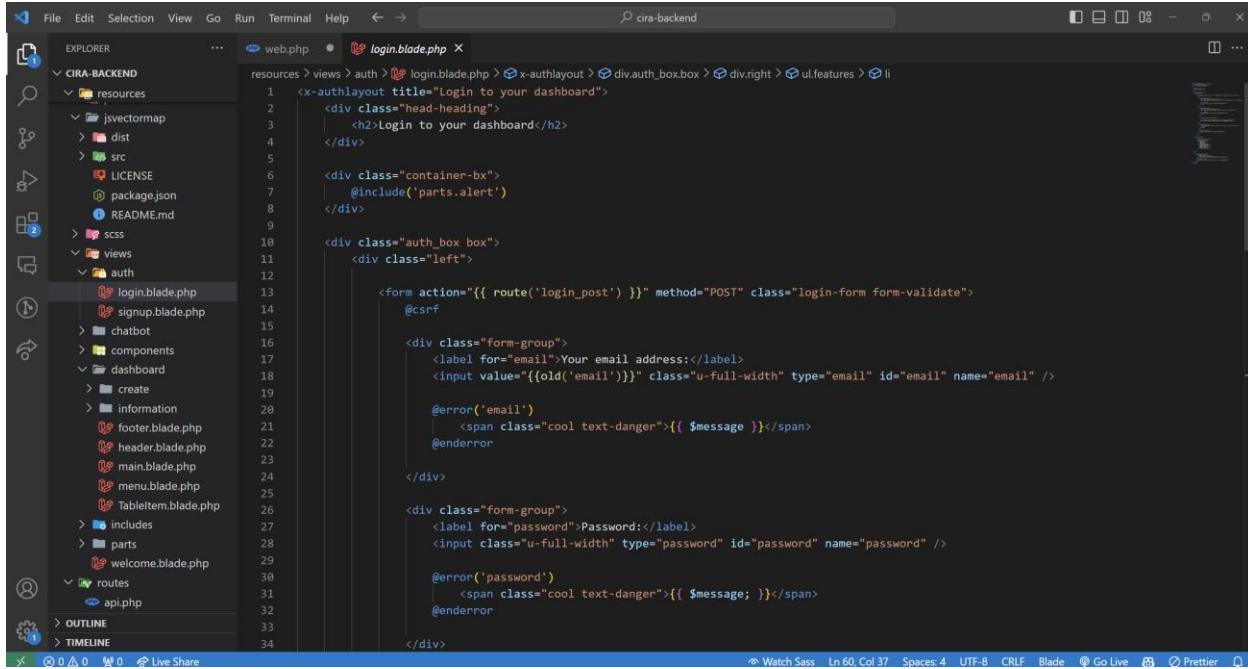
Powered By CIRA

9. SOURCE CODE

All the urls used in the page can be found in the routes/web.php file in Laravel.

9.1. LOGIN PAGE

The is the main code to render the login page. The request first is captured in the routes file then sent to the controller. The controller then renders the html file. When user click on login button it sends a post request to the same url but the POST request is connected to different POST controller the controller handles the user authentication process.



```
<> auth.layout title="Login to your dashboard">
    <div class="head-heading">
        <h2>Login to your dashboard</h2>
    </div>

    <div class="container-bx">
        @include('parts.alert')
    </div>

    <div class="auth_box box">
        <div class="left">
            <form action="{{ route('login_post') }}" method="POST" class="login-form form-validate">
                @csrf

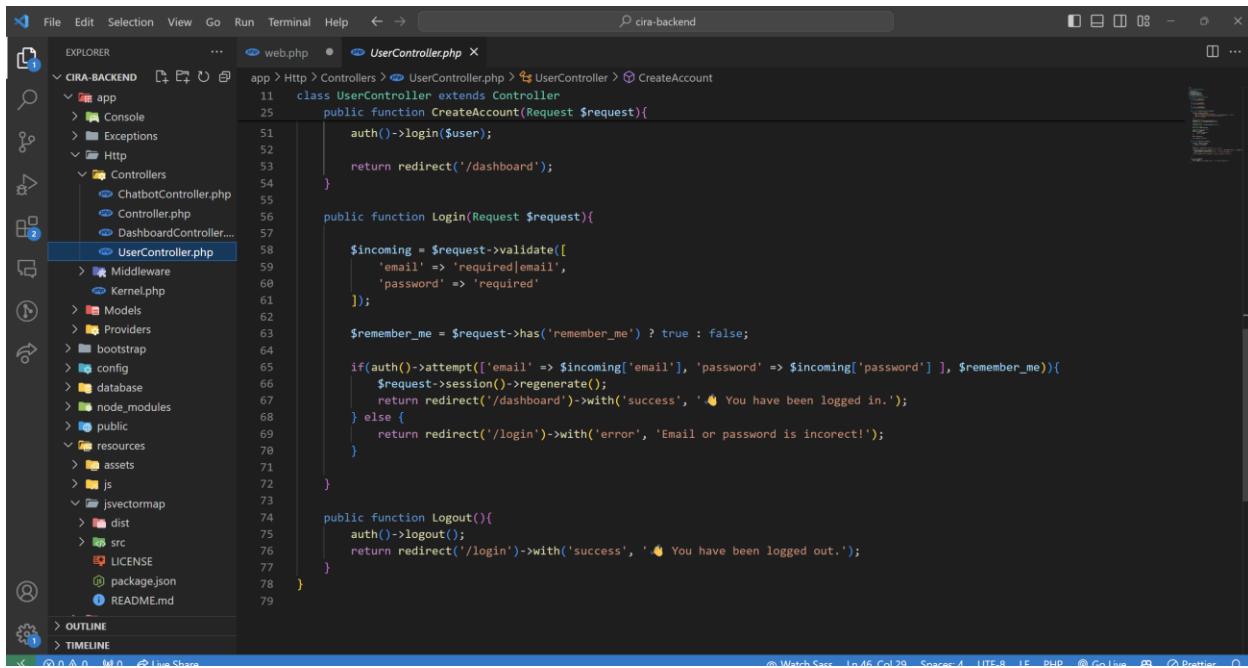
                <div class="form-group">
                    <label for="email">Your email address:</label>
                    <input value="{{ old('email') }}" class="u-full-width type="email" id="email" name="email" />

                    @error('email')
                        <span class="cool text-danger">{{ $message }}</span>
                    @enderror
                </div>

                <div class="form-group">
                    <label for="password">Password:</label>
                    <input class="u-full-width type="password" id="password" name="password" />

                    @error('password')
                        <span class="cool text-danger">{{ $message }}</span>
                    @enderror
                </div>
            </form>
        </div>
    </div>
```

The below login controller code handles the POST login code which redirects users to dashboard if their email and password is correct else shows an error message.



```
class UserController extends Controller
{
    public function CreateAccount(Request $request){
        auth()->login($user);

        return redirect('/dashboard');
    }

    public function Login(Request $request){

        $incoming = $request->validate([
            'email' => 'required|email',
            'password' => 'required'
        ]);

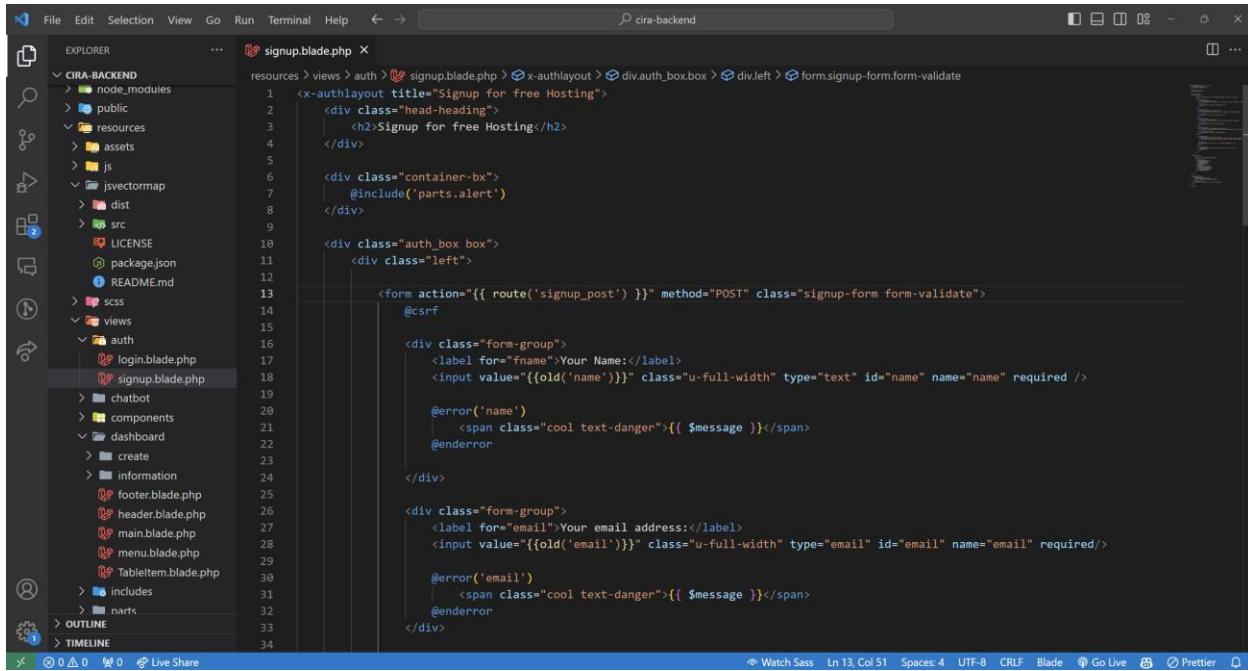
        $remember_me = $request->has('remember_me') ? true : false;

        if(auth()->attempt(['email' => $incoming['email'], 'password' => $incoming['password']], $remember_me)){
            $request->session()->regenerate();
            return redirect('/dashboard')->with('success', 'You have been logged in.');
        } else {
            return redirect('/login')->with('error', 'Email or password is incorrect!');
        }
    }

    public function Logout(){
        auth()->logout();
        return redirect('/login')->with('success', 'You have been logged out.');
    }
}
```

9.2. SIGNUP PAGE

The is the main code to render the signup page. The request first is captured in the routes file then sent to the controller. The controller then renders the html file. When user click on signup button it sends a post request to the same url but the POST request is connected to different POST controller the controller handles the user creation process.

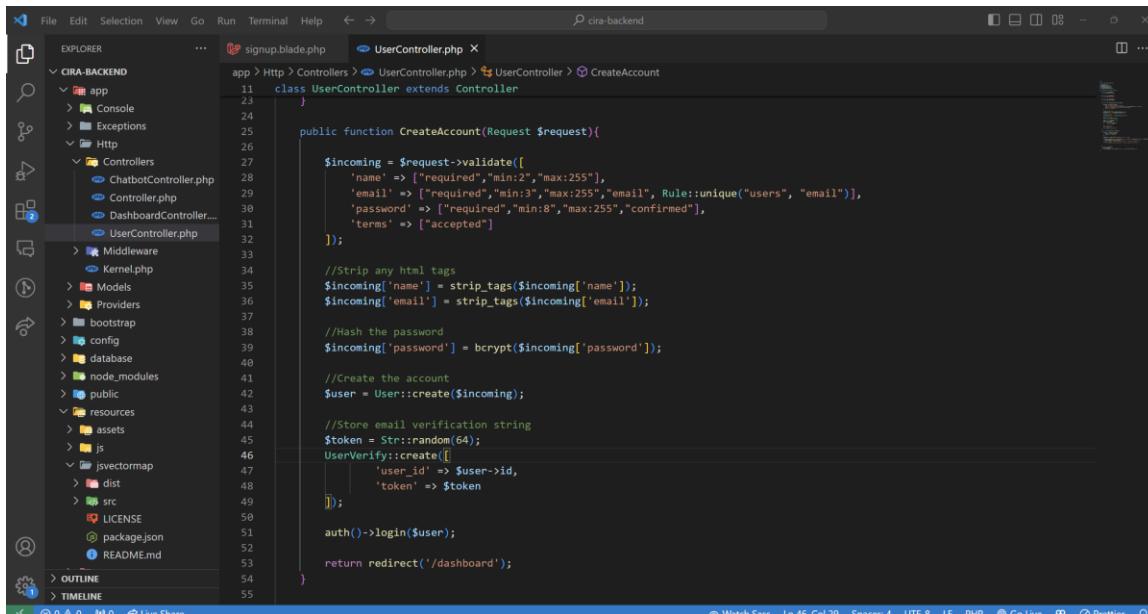


```
File Edit Selection View Go Run Terminal Help ↵ → cira-backend

EXPLORER ... signup.blade.php
CIRA-BACKEND resources > views > auth > signup.blade.php > x-authlayout > div.auth_box.box > div.left > form.signup-form.form-validate
1   <x-authlayout title="Signup for free Hosting">
2     <div class="head-heading">
3       <h2>Signup for free Hosting</h2>
4     </div>
5
6     <div class="container-bx">
7       @include('parts.alert')
8     </div>
9
10    <div class="auth_box box">
11      <div class="left">
12
13        <form action="{{ route('signup_post') }}" method="POST" class="signup-form form-validate">
14          @csrf
15
16          <div class="form-group">
17            <label for="name">Your Name:</label>
18            <input value="{{ old('name') }}" class="u-full-width type="text" id="name" name="name" required />
19
20            @error('name')
21              <span class="cool text-danger">{{ $message }}</span>
22            @enderror
23
24          </div>
25
26          <div class="form-group">
27            <label for="email">Your email address:</label>
28            <input value="{{ old('email') }}" class="u-full-width type="email" id="email" name="email" required />
29
30            @error('email')
31              <span class="cool text-danger">{{ $message }}</span>
32            @enderror
33
34        </form>
35
36      </div>
37    </div>
38
39  </div>
40
41  </div>
42
43  </div>
44
45  </div>
46
47  </div>
48
49  </div>
50
51  </div>
52
53  </div>
54
55

@ Watch Sass Ln 13, Col 51 Spaces: 4 UTF-8 CRLF Blade ⚡ Go Live ⚡ Prettier ⚡
```

The below code shows how the user account is created when user presses the Signup button in the signup page. It checks if email already exists or not, password and the confirmation matches then strips the data of any html codes so that we do not end up with sql injection. If everything is ok, we then create the account.



```
File Edit Selection View Go Run Terminal Help ↵ → cira-backend

EXPLORER ... UserController.php
CIRA-BACKEND app > Http > Controllers > UserController.php > UserController > CreateAccount
11  class UserController extends Controller
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

class UserController extends Controller
{
    public function CreateAccount(Request $request)
    {
        $incoming = $request->validate([
            'name' => ['required', 'min:2', 'max:255'],
            'email' => ['required', 'min:3', 'max:255', 'email', Rule::unique("users", "email")],
            'password' => ['required', 'min:8', 'max:255', 'confirmed'],
            'terms' => ['accepted']
        ]);

        //Strip any html tags
        $incoming['name'] = strip_tags($incoming['name']);
        $incoming['email'] = strip_tags($incoming['email']);

        //Hash the password
        $incoming['password'] = bcrypt($incoming['password']);

        //Create the account
        $user = User::create($incoming);

        //Store email verification string
        $token = Str::random(64);
        UserVerify::create([
            'user_id' => $user->id,
            'token' => $token
        ]);

        auth()->login($user);

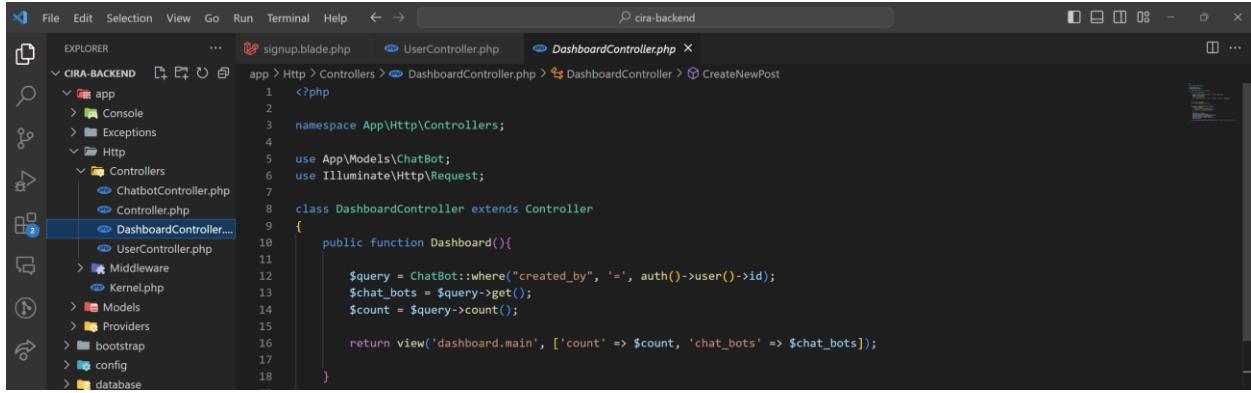
        return redirect('/dashboard');
    }
}

@ Watch Sass Ln 46, Col 29 Spaces: 4 UTF-8 LF PHP ⚡ Go Live ⚡ Prettier ⚡
```

9.3. DASHBOARD PAGE

The main component of the dashboard page is the Chatbot listing. This is the code to render the chatbot lists in the page.

This simple piece of controller code connects to the database and fetches the user's chatbots and sends it to the view to render.

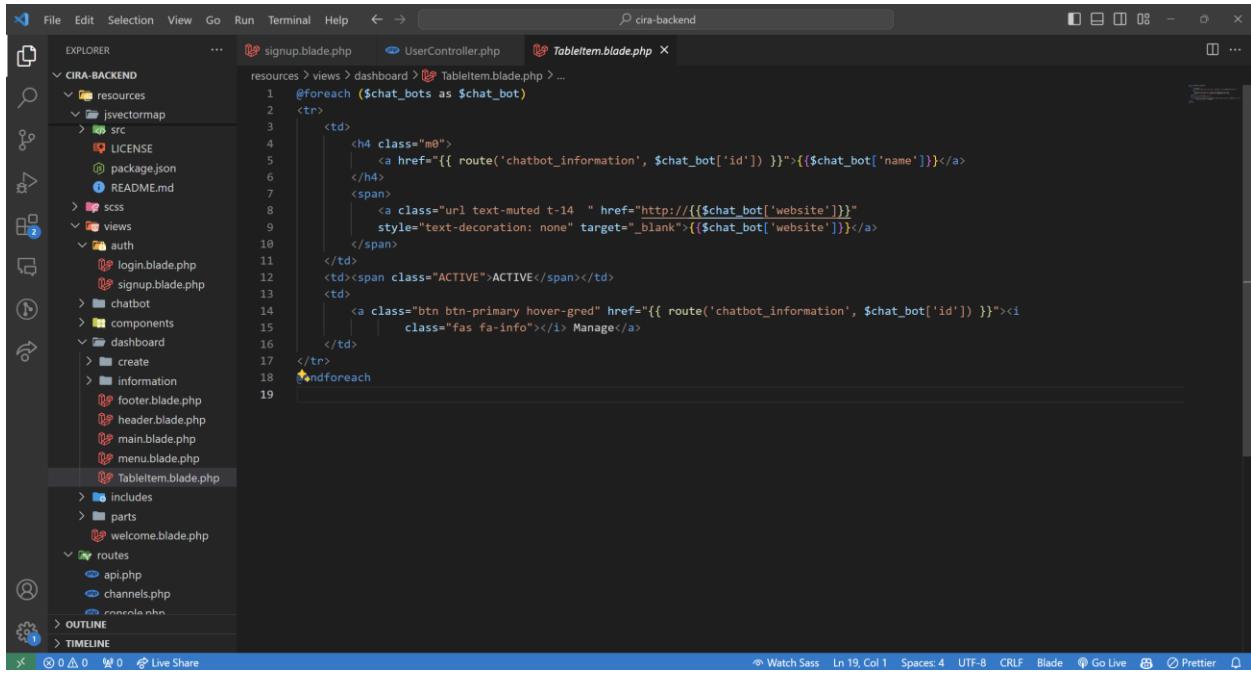


A screenshot of a code editor showing the `DashboardController.php` file. The code defines a `DashboardController` class that extends `Controller`. It contains a `Dashboard()` method that performs a database query to find chatbots created by the authenticated user and returns a view with the count and a list of chatbots.

```
<?php
namespace App\Http\Controllers;
use App\Models\ChatBot;
use Illuminate\Http\Request;
class DashboardController extends Controller
{
    public function Dashboard(){
        $query = ChatBot::where("created_by", '=', auth()->user()->id);
        $chat_bots = $query->get();
        $count = $query->count();

        return view('dashboard.main', ['count' => $count, 'chat_bots' => $chat_bots]);
    }
}
```

We loop through all the sent data and render each item as a table row so that the chatbots are rendered clearly and correctly.



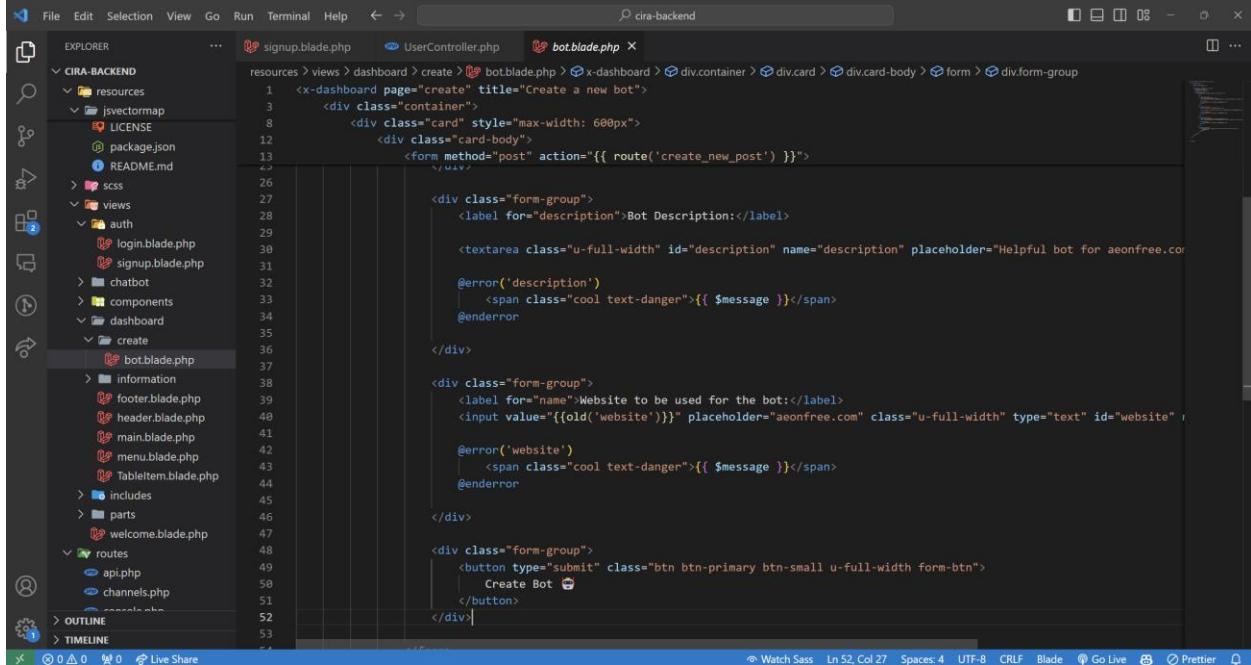
A screenshot of a code editor showing the `TableItem.blade.php` file. The code uses a `@foreach` loop to iterate over `$chat_bots`. For each bot, it creates a table row (`<tr>`) containing a name link (`{{ \$chat_bot\['name'\] }}`), a website link (`{{ \$chat_bot\['website'\] }}`), and a manage button (`Manage`).

```
@foreach ($chat_bots as $chat_bot)
|  |  |  |
| --- | --- | --- |
| {{ \$chat\_bot\['name'\] }}  {{ \$chat\_bot\['website'\] }} | {{ $chat_bot['website'] }} | Manage |

@endforeach
```

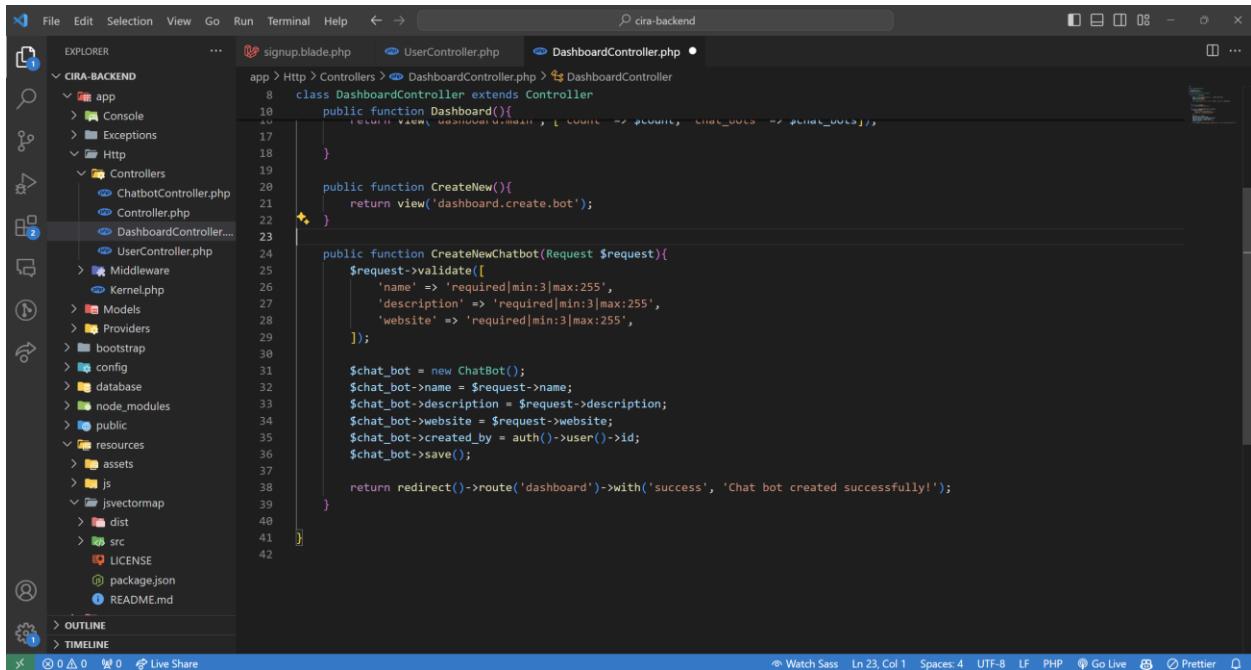
9.3. CREATE CHATBOT

The below view renders the chatbot creation page. User can add the necessary info and click on Create Bot button to create their chatbot. The form will send a post request to the required route and starts the chatbot creation process.



```
resources > views > dashboard > create > bot.blade.php > x-dashboard > div.container > div.card > div.card-body > form > div.form-group
1   <x-dashboard page="create" title="Create a new bot">
3     <div class="container">
5       <div class="card" style="max-width: 600px;">
7         <div class="card-body">
9           <form method="post" action="{{ route('create_new_bot') }}>
11             <div class="form-group">
13               <label for="description">Bot Description:</label>
15               <textarea class="u-full-width" id="description" name="description" placeholder="Helpful bot for aeonfree.com" type="text">
17               @error('description')
19                 <span class="cool text-danger">{{ $message }}</span>
20               @enderror
22             </div>
24             <div class="form-group">
26               <label for="name">Website to be used for the bot:</label>
28               <input value="{{ old('website') }}" placeholder="aeonfree.com" class="u-full-width" type="text" id="website">
30               @error('website')
32                 <span class="cool text-danger">{{ $message }}</span>
33               @enderror
35             </div>
37             <div class="form-group">
39               <button type="submit" class="btn btn-primary btn-small u-full-width form-btn">
41                 Create Bot
42               </button>
44             </div>
46           </form>
48         </div>
50       </div>
52     </div>
54   </x-dashboard>
```

The below controller code will check all the required information of the chatbot and create the chatbot for the user.

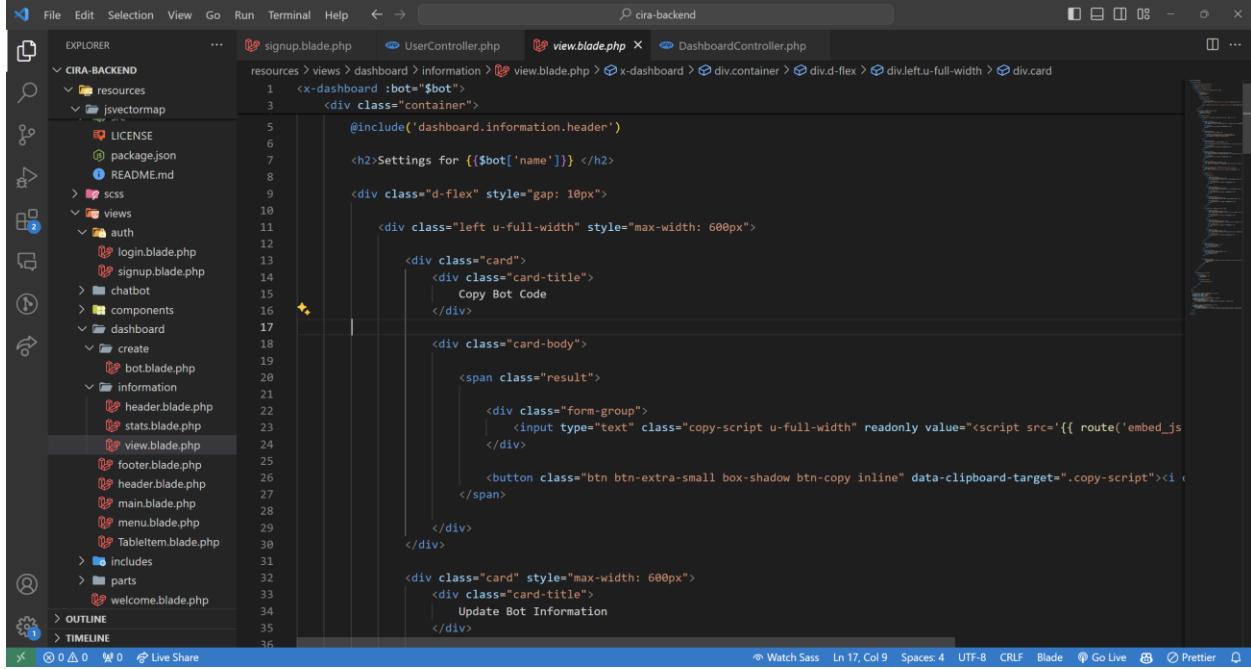


```
app > Http > Controllers > DashboardController.php > DashboardController
8   class DashboardController extends Controller
10   public function Dashboard(){
11     return view('dashboard.main', [
12       'count' => $count,
13       'chat_docs' => $chat_docs
14     ]);
15   }
17   public function CreateNew(){
18     return view('dashboard.create.bot');
19   }
21   public function CreateNewChatbot(Request $request){
22     $request->validate([
23       'name' => 'required|min:3|max:255',
24       'description' => 'required|min:3|max:255',
25       'website' => 'required|min:3|max:255',
26     ]);
27
28     $chat_bot = new ChatBot();
29     $chat_bot->name = $request->name;
30     $chat_bot->description = $request->description;
31     $chat_bot->website = $request->website;
32     $chat_bot->created_by = auth()->user()->id;
33     $chat_bot->save();
34
35     return redirect()->route('dashboard')->with('success', 'Chat bot created successfully!');
36   }
37 }
```

The url is then sent to the queue for data scraping.

9.4. MANAGE CHATBOT

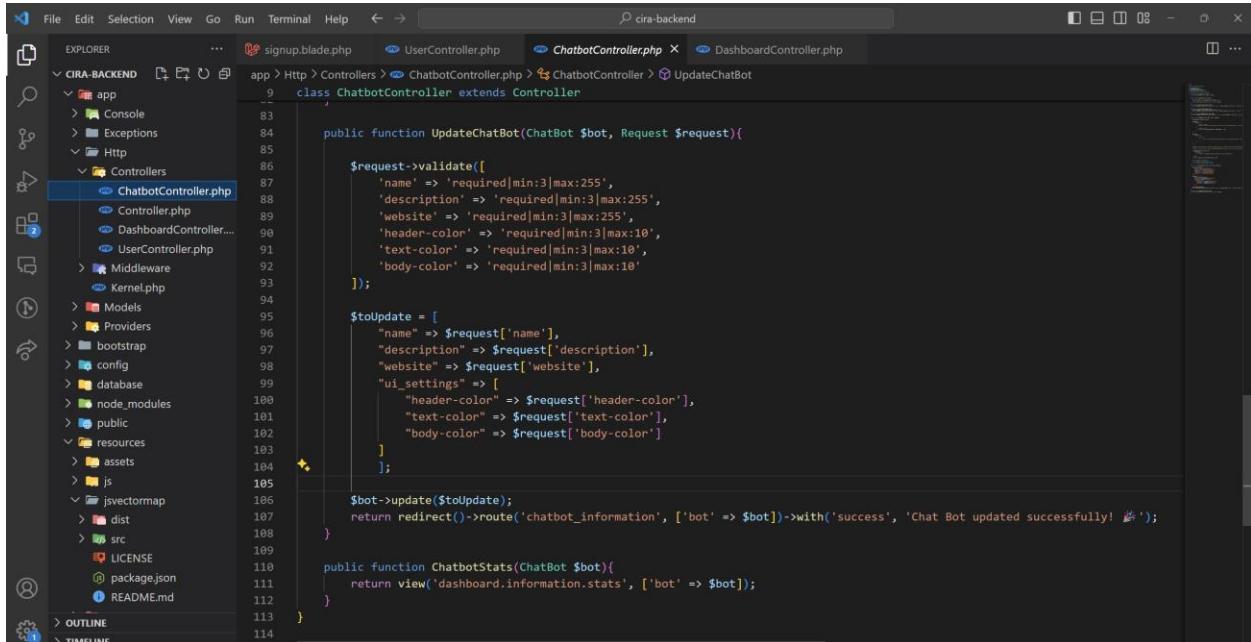
The interface to manage the chatbot, user can customize everything they need from this page.



```
resources > views > dashboard > information > view.blade.php > x-dashboard > div.container > div.d-flex > div.left.u-full-width > div.card
1   <x-dashboard :bot="$bot">
3     <div class="container">
5       @include('dashboard.information.header')
6
7       <h2>Settings for {$bot['name']} </h2>
8
9       <div class="d-flex" style="gap: 10px">
10         <div class="left u-full-width" style="max-width: 600px">
11           <div class="card">
12             <div class="card-title">
13               Copy Bot Code
14             </div>
15
16             <div class="card-body">
17               <span class="result">
18                 <div class="form-group">
19                   <input type="text" class="copy-script u-full-width" readonly value="{{ route('embed_js', $bot) }}"
20                 </div>
21
22                 <button class="btn btn-extra-small box-shadow btn-copy inline" data-clipboard-target=".copy-script"><i class="fa fa-clone" /> Copy
23               </span>
24             </div>
25
26             <div class="card" style="max-width: 600px">
27               <div class="card-title">
28                 Update Bot Information
29               </div>
30             </div>
31
32             <div class="card" style="max-width: 600px">
33               <div class="card-title">
34                 <button class="btn btn-extra-small box-shadow" type="button" data-bs-dismiss="modal" data-bs-target="#updateBotModal">Cancel
35               </div>
36             </div>

```

The updated values are then processed through the controller.

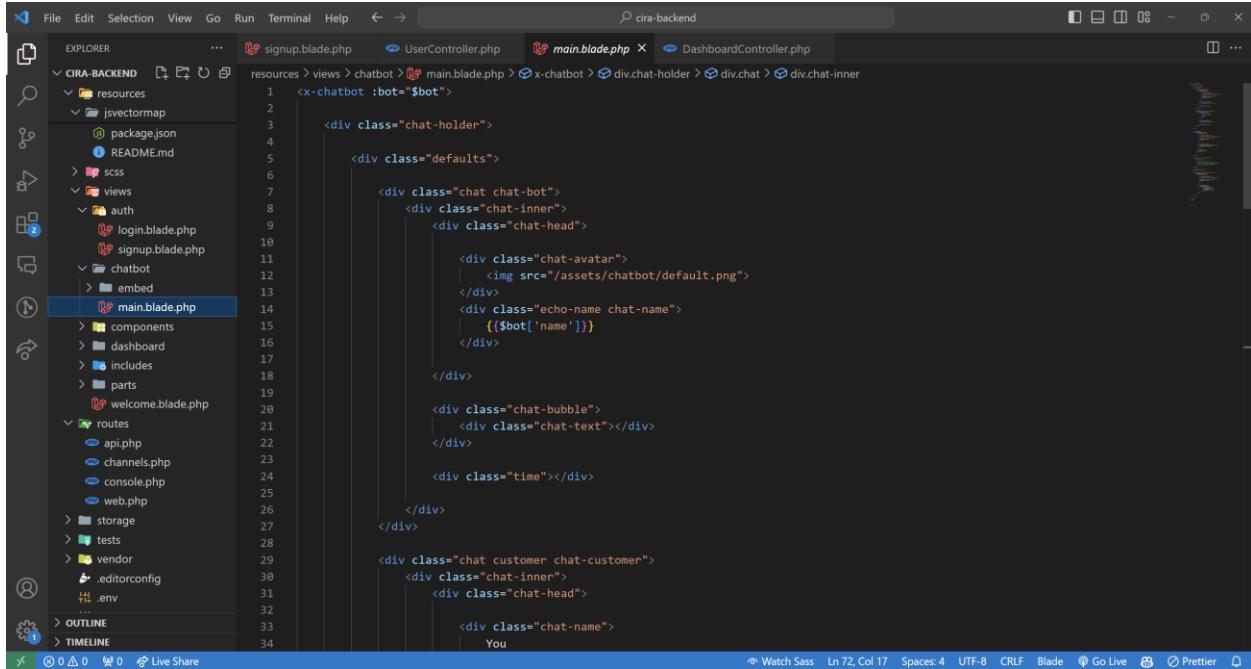


```
app > Http > Controllers > ChatbotController.php > ChatbotController > UpdateChatBot
9  class ChatbotController extends Controller
10
11  public function UpdateChatBot(ChatBot $bot, Request $request){
12
13    $request->validate([
14      'name' => 'required|min:3|max:255',
15      'description' => 'required|min:3|max:255',
16      'website' => 'required|min:3|max:255',
17      'header-color' => 'required|min:3|max:10',
18      'text-color' => 'required|min:3|max:10',
19      'body-color' => 'required|min:3|max:10'
20    ]);
21
22    $toUpdate = [
23      "name" => $request['name'],
24      "description" => $request['description'],
25      "website" => $request['website'],
26      "ui_settings" => [
27        "header-color" => $request['header-color'],
28        "text-color" => $request['text-color'],
29        "body-color" => $request['body-color']
30      ]
31    ];
32
33    $bot->update($toUpdate);
34    return redirect()->route('chatbot_information', ['bot' => $bot])->with('success', 'Chat Bot updated successfully! 🎉');
35  }
36
37  public function ChatbotStats(ChatBot $bot){
38    return view("dashboard.information.stats", ['bot' => $bot]);
39  }

```

9.5. CHATBOT DESIGN

This just a hollow empty html page to hold the chatbot's information. The data is passed through the controller and rendered correctly. Some of the functions like receiving messages are rendered through javascript.

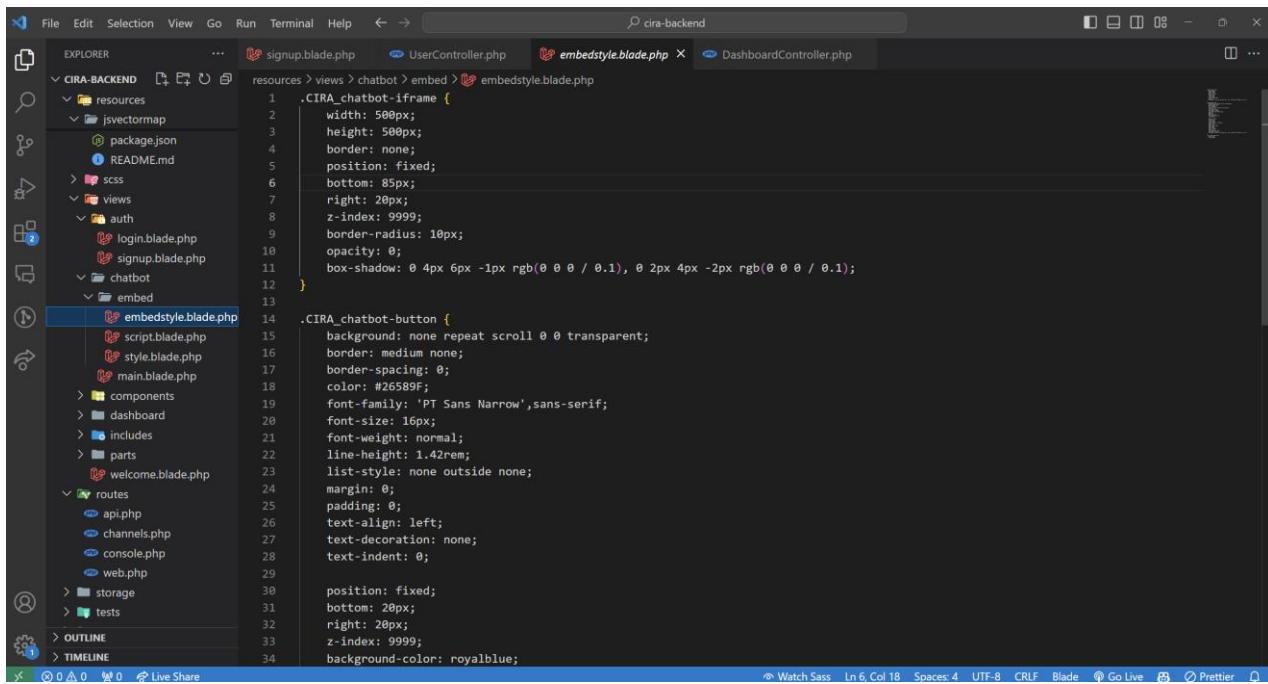


The screenshot shows the VS Code interface with the 'cira-backend' workspace open. The Explorer sidebar on the left shows the project structure, including 'CIRA-BACKEND', 'resources', 'views', 'auth', 'embed', and 'routes'. The main editor area displays the 'main.blade.php' file, which contains the following Blade template code:

```
<x>chatbot :bot="$bot">
    <div class="chat-holder">
        <div class="defaults">
            <div class="chat chat-bot">
                <div class="chat-inner">
                    <div class="chat-head">
                        <div class="chat-avatar">
                            
                        </div>
                        <div class="echo-name chat-name">
                            {{ $bot['name'] }}
                        </div>
                    </div>
                    <div class="chat-bubble">
                        <div class="chat-text"></div>
                    </div>
                    <div class="time"></div>
                </div>
            <div class="chat customer chat-customer">
                <div class="chat-inner">
                    <div class="chat-head">
                        <div class="chat-name">
                            You
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</x>
```

The status bar at the bottom indicates 'Watch Sass' and 'Ln 72, Col 17'.

This embed file is technically a blade file but we render it as a css so that we can use the user values from the database in the css code. This way the design created by the user is rendered correctly.



The screenshot shows the VS Code interface with the 'cira-backend' workspace open. The Explorer sidebar on the left shows the project structure, including 'CIRA-BACKEND', 'resources', 'views', 'auth', 'embed', and 'routes'. The main editor area displays the 'embedstyle.blade.php' file, which contains the following Blade template code:

```
.CIRA_chatbot-iframe {
    width: 500px;
    height: 500px;
    border: none;
    position: fixed;
    bottom: 85px;
    right: 20px;
    z-index: 9999;
    border-radius: 10px;
    opacity: 0;
    box-shadow: 0 4px 6px -3px rgba(0 0 0 / 0.1), 0 2px 4px -2px rgba(0 0 0 / 0.1);
}

.CIRA_chatbot-button {
    background: none repeat scroll 0 0 transparent;
    border: medium none;
    border-spacing: 0;
    color: #265B9F;
    font-family: 'PT Sans Narrow', sans-serif;
    font-size: 16px;
    font-weight: normal;
    line-height: 1.42rem;
    list-style: none outside none;
    margin: 0;
    padding: 0;
    text-align: left;
    text-decoration: none;
    text-indent: 0;
    position: fixed;
    bottom: 20px;
    right: 20px;
    z-index: 9999;
    background-color: royalblue;
```

The status bar at the bottom indicates 'Watch Sass' and 'Ln 6, Col 18'.

9.6. CHATBOT LOGIC

The responses send to the CIRA frontend is generated through this piece of code. It uses a custom api created on a python backend which interacts with open source LLM from Meta. We send the user generated questions and our default prompts and it return us answers. Those answers are then sent to the client side as responses.

```

    public function BotRequest(ChatBot $bot, Request $request) {
        $userMessage = $request['message'];

        $messages = [
            'messages' => [
                [
                    'role' => "system",
                    "content" => "You are an AI Chat Bot who answers all of the customer questions. In every answers use expressive emojis. Make answers short and precise use professional words. Don't mention"
                ],
                [
                    'role' => "user",
                    "content" => isset($userMessage) ? $userMessage : "Hello!"
                ]
            ]
        ];

        $message = [
            'contents' => [
                [
                    'parts' => [
                        [
                            'text' => "You are an AI Chat Bot who answers all of the customer questions. CIRA is an organization who built this chat bot. Do not provide any coding assistance not related to thi"
                        ]
                    ]
                ]
            ]
        ];
    }

    $response = Http::post("http://localhost:8998/api/generateResponse", $message);

    // $response = Http::withToken("M-7E3LECPbfPmC12wKsk2tch-fTmhouse")->post('https://api.cloudflare.com/client/v4/accounts/b74a406241cf186d14c18dcf94f26f/al/run@cf/meta/llama-2-7b-chat-int8', $message);
    if(isset($response['candidates'])) {
        $result = [
            'result' => [
                'response' => $response['candidates'][0]['content'][0]['text']
            ]
        ];
    } else {
        return response()->json($response, 500);
    }
    return response()->json($result);
    return response()->json($response->json());
}

```

The below code generates the user designed custom javascript codes and css codes to improve the overall experience of the chatbot.

```

    use App\Models\Chatbot;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;

class ChatbotController extends Controller
{
    public function ChatBot(ChatBot $bot){
        return view('chatbot.main', ['bot' => $bot]);
    }

    public function ViewChatBot(ChatBot $bot){

        $bot->ui_settings = json_decode($bot->ui_settings, true);

        return view('dashboard.information.view', ['bot' => $bot]);
    }

    public function GenerateScript(ChatBot $bot){
        return response()->view('chatbot.embed.script', ['bot' => $bot])->header('Content-Type', 'text/javascript');
    }

    public function GenerateStyle(ChatBot $bot){
        $bot->ui_settings = json_decode($bot->ui_settings, true);
        return response()->view('chatbot.embed.style', ['bot' => $bot])->header('Content-Type', 'text/css');
    }

    public function GenerateEmbedStyle(ChatBot $bot){
        return response()->view('chatbot.embed.embedstyle', ['bot' => $bot])->header('Content-Type', 'text/css');
    }

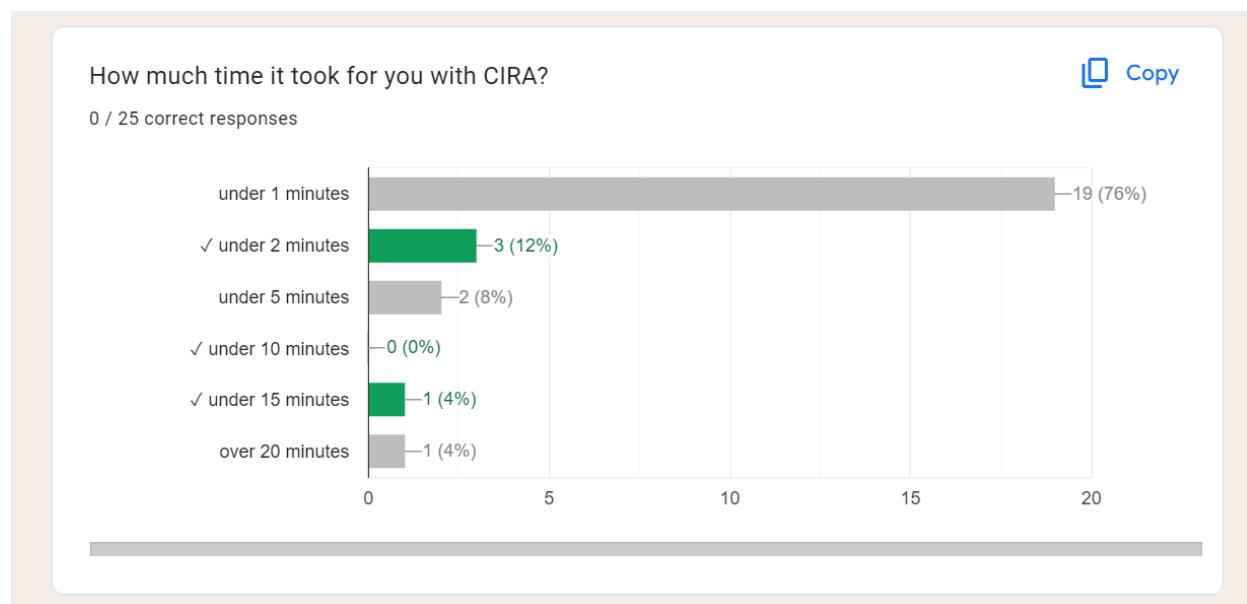
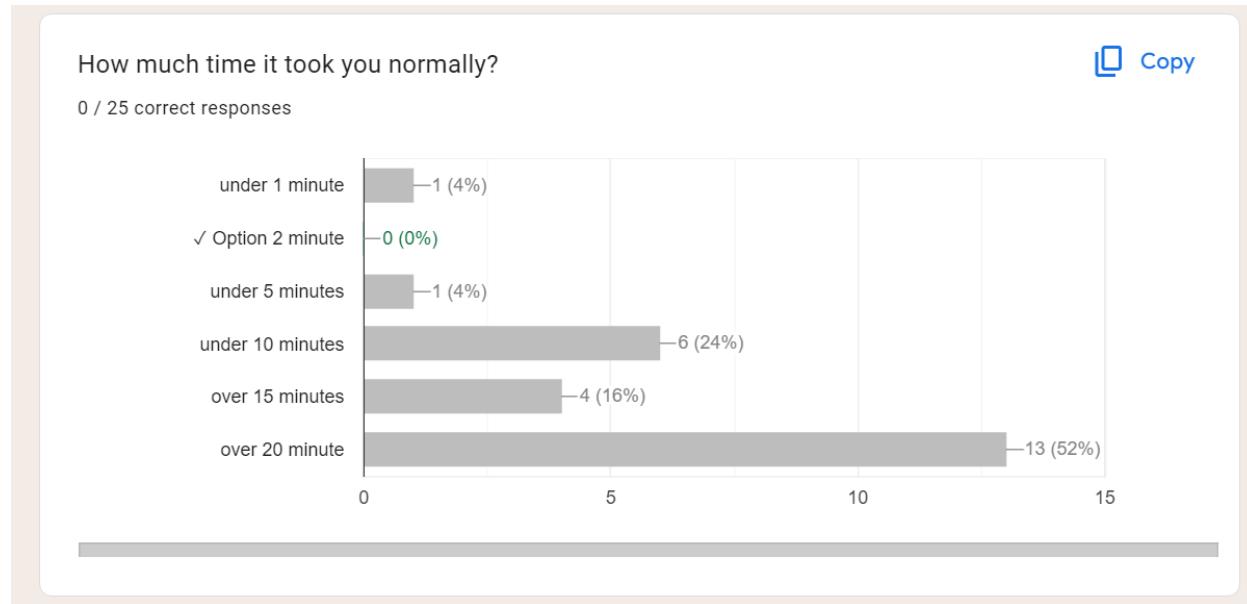
    public function BotRequest(ChatBot $bot, Request $request){
        $userMessage = $request['message'];
    }
}

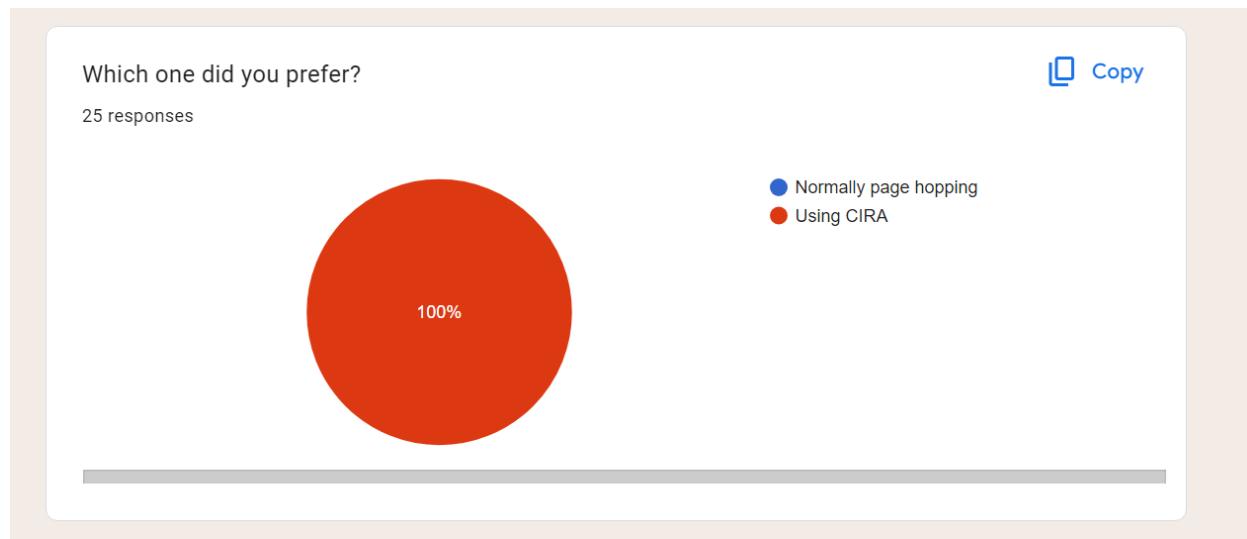
```

10. RESULTS AND OBSERVATIONS

We conducted a survey to gather insights on the efficiency of finding specific information and preferences among students at LPU, the objective of this survey was to understand the time it takes for participants to find specific information such as a per credit fee for data analytics for M.Sc. Data Science from The website of University of Texas at Arlington and also the preference between traditional methodology of finding information using page hopping and using CIRA.

In this survey we found out that 11 students took more than 5 minutes to find the information using normal method and 13 students took a whooping over 20 minutes to find this information whereas with CIRA, an astonishing 19 (76%) of the participants found the information under one minute with the help of CIRA. Also, 100% of participants preferred using CIRA to search for information in the websites.





11. CIRA DEPLOYMENT

The backend of CIRA is created with Laravel so, it must be hosted with a cloud provider. In this case, CIRA was deployed to a Cloud hosting service with the help of a VPS.

To analyze the use case of the chatbot, CIRA was deployed to a free hosting service called AeonFree. It has a huge library of documentation and user guides on multiple pages, those links were perfectly scraped with the help of CIRA. CIRA Chatbot was added to AeonFree for 15 days and the user content level, number of questions asked through CIRA, and if their issue was resolved with respect to the countries were analyzed.

User Content Level with respect to countries.



Figure: User Content level for the responses provided by the chatbot, Users of the chatbot could like or dislike the responses.

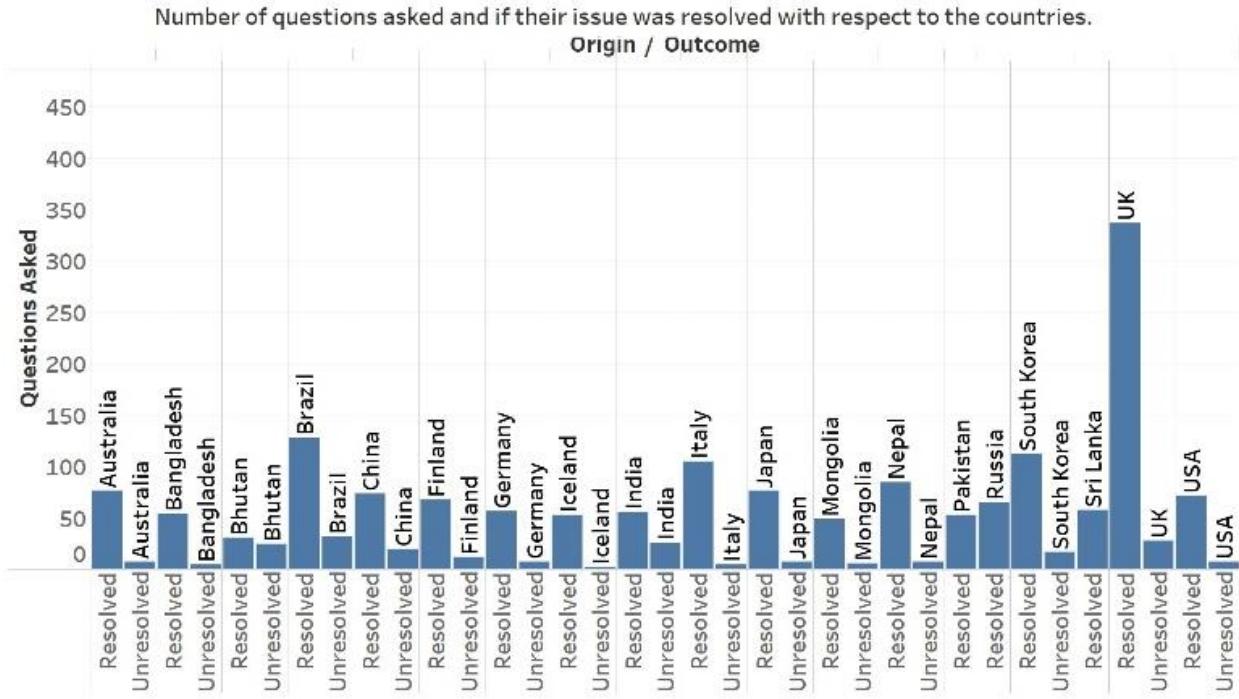


Figure: Number of questions asked through CIRA and if their issue was resolved with respect to the countries.

From the above figure, It can be realized that more than 85% of the users of CIRA were happy with the level of answers provided similarly in another graph, in the above figure, it can be seen that user's most of the queries were resolved, also the user seems to be engaging with CIRA with an average number of questions asked per country at 200 in just 15 days of its deployment even though no such announcement was made that CIRA was an LNLM based chatbot. Though it cannot be shown through data because of privacy, the users seem to be asking a lot of personal questions and resolving their everyday matters through the help of CIRA just like it has been found in other LLMs like ChatGPT, Gemini, Mistral, and Grok, etc.

11. CONCLUSION

CIRA makes the information retrieval process smooth, enhances user engagement and provides personal assistance, by leveraging the Large Language Model (LLM) LLaMA2. CIRA represents a milestone in the field of AI driven chatbot technology for a diverse field. This research paper traverses through the evolution of chatbot technology like GALGOBOT, Maya, and Jollity Chatbot to understand the significance of user centric design and adaptability in crafting chatbot interfaces. From prototype to development, CIRA was handcrafted to be easy to use by users and easy to deploy by website owners while not missing the most important AI landscape. Website owners can easily create and customize the chatbot without much effort and analyze user interactions with a

breeze. This project aims to stand as a beacon of innovation, by announcing a new era of intelligent virtual assistants to revolutionize digital interactions.

11. PUBLICATION DETAILS

 Gmail Sameer Khanal <sameerkhanal2099@gmail.com>

Fwd: Paper Acceptance | ICONIC-2024

Sumit Paudel <paudel669@gmail.com>
To: Sameer Khanal <sameerkhanal2099@gmail.com> Fri, May 3, 2024 at 1:47 AM

----- Forwarded message -----
From: Ash <sheldonashim37@gmail.com>
Date: Mon, Apr 15, 2024 at 4:47 PM
Subject: Fwd: Paper Acceptance | ICONIC-2024
To: Sumit Paudel <paudel669@gmail.com>

----- Forwarded message -----
From: Iconic . <iconic@ipu.co.in>
Date: Mon, 15 Apr 2024, 11:38 am
Subject: Paper Acceptance | ICONIC-2024
To: Ash <sheldonashim37@gmail.com>

Dear Author

Greetings from ICONIC-2024!!

Congratulations, we are happy to inform you that your submitted manuscript, identified as **ICONIC-078**, has been **ACCEPTED** for the final presentation in ICONIC-2024 subject to the incorporation of attached comments from the review committee.

Please review the comments to be incorporated into the camera-ready paper. The revised paper should be submitted by **April 17th i.e. Wednesday**.

Reviewer I

1. Avoid abbreviations in title, Improve the abstract, Improve the quality of the figure
2. Reference to figures not given in text or paragraphs anywhere.

Reviewer II

1. Literature review may be presented in tabular form with pros and cons of reviewed methods.
2. Quality of figure to be improved so that text can be easily read.

To confirm your participation kindly complete the registration process by depositing the Conference Fee. Payment Link: [Online Registration - ICONIC 2024 \(ipu.in\)](#)

Kindly reply to this email (**NO FRESH EMAIL**) to submit the following documents.

1. Final Camera-Ready Copy (CRC) as per the IEEE format. The final file should not have plagiarism more than 10% [**File should be in editable mode**]
2. Copy of e-receipt of registration fees.

Wishing you the best!!

--

Regards
Team ICONIC- 2024
ICONIC
Conference Website : <https://conferences.ipu.in/iconic/>

12. AWARD

Our paper was also selected as the best among 216 submissions made in the ICONIC conference held by IBM supported by IEEE. We were awarded as “The best paper of ICONIC 2024”.



13. REFERENCES

- [1] A. Vaswani et al., “Attention Is All You Need,” Jun. 2017, [Online]. Available: <http://arxiv.org/abs/1706.03762>.
- [2] A. Følstad and M. Skjuve, “Chatbots for customer service: User experience and motivation,” in ACM International Conference Proceeding Series, Association for Computing Machinery, Aug. 2019.
- [3] E. Adamopoulou and L. Moussiades, “An Overview of Chatbot Technology,” 2020, pp. 373–383.
- [4] S. Saraswat, S. Mishra, V. Mani, and S. Priya, “GALGOBOT - The college companion chatbot,” in Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 1374–1378.
- [5] I. Medhi Thies, N. Menon, S. Magapu, M. Subramony, and J. O’Neill, “How do you want your chatbot? An exploratory Wizard-of-Oz study with young, Urban Indians,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2017, pp. 441–459.
- [6] K. Deepika, V. Tilekya, J. Mamatha, and T. Subetha, “Jollity Chatbot- A contextual AI Assistant,” in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 1196–1200.
- [7] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, “Challenges in Chatbot Development: A Study of Stack Overflow Posts,” in Proceedings - 2020 IEEE/ACM 17th International Conference on Mining Software Repositories, MSR 2020, Association for Computing Machinery, Inc, Jun. 2020, pp. 174–185.
- [8] W. Ye and Q. Li, “Chatbot Security and Privacy in the Age of Personal Assistants,” in Proceedings - 2020 IEEE/ACM Symposium on Edge Computing, SEC 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 388–393.
- [9] G. Santoso, J. Setiawan, and A. Sulaiman, “Development of OpenAI API Based Chatbot to Improve User Interaction on the JBMS Website,” G-Tech: Jurnal Teknologi Terapan, vol. 7, no. 4, pp. 1606–1615, Oct. 2023.

14. PLAGIARISM REPORT

Contextual Information Retrieval Assistant (CIRA) Powered by
Local Neural Language Model (LNLM)

ORIGINALITY REPORT



PRIMARY SOURCES

1	www.slideshare.net Internet Source	1 %
2	www.coursehero.com Internet Source	1 %
3	core.ac.uk Internet Source	1 %
4	www.semanticscholar.org Internet Source	<1 %
5	onlinelibrary.wiley.com Internet Source	<1 %
6	technodocbox.com Internet Source	<1 %
7	Submitted to Champlain College Student Paper	<1 %
8	www.scribd.com Internet Source	<1 %
9	Submitted to Liverpool John Moores University	<1 %

10	Submitted to South Bank University Student Paper	<1 %
11	www.passeidireto.com Internet Source	<1 %
12	resources.finalsuite.net Internet Source	<1 %
13	Zhirun Zhang, Xinzhi Zhang, Li Chen. "Informing the Design of a News Chatbot", Proceedings of the 21th ACM International Conference on Intelligent Virtual Agents, 2021 Publication	<1 %
14	Shruti Patil, Venkatesh M. Mudaliar, Pooja Kamat, Shilpa Gite. "LSTM based Ensemble Network to enhance the learning of long-term dependencies in chatbot", International Journal for Simulation and Multidisciplinary Design Optimization, 2020 Publication	<1 %
15	arxiv.org Internet Source	<1 %
16	ejournal.uniramalang.ac.id Internet Source	<1 %
17	www.mdpi.com Internet Source	<1 %
	ijrpr.com	

18

Internet Source

<1 %

Exclude quotes Off
Exclude bibliography On

Exclude matches Off

The plagiarism report was checked by our mentor. The Log id of the report is: 16160