

REPORT ON  
**BLOCKGPT – CHATBOT FOR BLOCKCHAIN DOMAIN**

By:

Name: Enrollment No.  
Dharmanshu Singh 210398

Mentored By:

Dr. Soharab Hossain

*Prepared in the partial fulfillment of the*

Practice School II Course

AT

**Dolf. Finance**

**Address:** 1<sup>st</sup> and 2<sup>nd</sup> Floor, Plus Offices, Landmark Cyber Park, Sector-67,  
Gurugram, India

**A Practice School II Station of**



# Certificate



## Certificate of authenticity

This is to certify that Practice School Project of **DHARMANSHU SINGH** titled **BLOCKGPT-CHATBOT FOR BLOCKCHAIN DOMAIN** to the best of my knowledge is a record of bonafide work carried out by him under my guidance and/or supervision. The contents embodied in this report, to the best of my knowledge have not been submitted anywhere else in any form for the award of any other degree or diploma. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried during **1st June 2023 to 31st July 2023** in Dolf. Finance.

### Signature of PS-II Faculty advisor

Name: Dr. Soharab Hossain

Designation: Associate Professor  
SOET-CSE

Date:

### Signature of Industry Mentor

A handwritten signature in blue ink that appears to read "Ravi Kumar".

Name: Mr. Ravi Chaurasia

Designation: CTO



09-August-2023  
Seal of Organization with date:

## Joining Report

### BML MUNJAL UNIVERSITY PRACTICE SCHOOL – II JOINING REPORT

<b>Name of the Student</b>	Dharmanshu Singh
<b>Name and Address of the Practice School – II Station</b>	Dolf. Finance, 1 <sup>st</sup> & 2 <sup>nd</sup> Floor, Landmark Cyber Park, Sector 67, Gurugram, Haryana-122101
<b>Date of Joining PS-II station as per offer letter</b>	29 <sup>th</sup> May 2023
<b>Actual date of reporting to PS-II station</b>	4 <sup>th</sup> July 2023
<b>Department Allocated</b>	Software Development
<b>Name and Designation of the Industry Guide/ Industry Mentor for the Project</b>	Ravi Chaurasia, CTO
<b>Industry Mentor Contact No. (If available)</b>	
<b>Industry Mentor E-mail Address (Compulsory)</b>	ravi@dolf.finance

## **Acknowledgements**

I would like to express my heartfelt gratitude to all the individuals and organizations who have contributed to guidance and successful completion of the project.

### **Head of Organization:**

I am deeply thankful for Prof. (Dr.) Maneek Kumar and Prof. Shyam Menon for this opportunity to work at this startup.

### **Academic advisors and Industry mentors:**

I am deeply thankful to Dr. Soharab Hossain and Mr. Ravi Chaurasia, who provided invaluable guidance and unwavering support throughout the course of the project. Their expertise and insights have been instrumental in successful completion of the project.

### **Colleagues:**

I extend my appreciation to Ayush Kumar, Sankalp Mane for their collaborative efforts and constructive feedback during the course of the project. Their valuable suggestions have greatly enriched the final outcome.

### **Coordinators of PS-II Programme:**

I would also like to thank Dr. Kiran Khatter, Dr. Ranbir Singh, and Dr. Yogesh Gupta for allowing us to pursue this opportunity.

## **Abstract**

Blockchain technology has changed many industries, including healthcare, finance, and gaming.. Yet, as the blockchain landscape grows in complexity, the need for proficient support and communication within this industry becomes increasingly evident. This initiative introduces BlockGPT, an advanced chatbot meticulously crafted to cater to the unique requirements of the blockchain field.

Utilizing the GPT-3.5 Architecture, BlockGPT harnesses the potential of machine learning to facilitate smooth interactions between users and the intricate mechanisms of blockchain technology. Leveraging Langchain LLM and trained on meticulously sourced data, BlockGPT possesses the ability to comprehensively comprehend and tackle an extensive array of inquiries and challenges related to blockchain.

To encapsulate, BlockGPT functions as a valuable virtual chatbot, facilitating seamless communication and assistance within the realm of blockchain. By harnessing the prowess of blockchain technology, it emerges as a pivotal instrument for all parties involved in this rapidly evolving tech domain. Its aptitude to comprehend and effectively resolve diverse blockchain inquiries positions it as an indispensable resource amidst the dynamic technological shifts.

## **Table of Contents**

<b>Contents</b>	<b>Page No.</b>
1. Introduction	01
2. Overview	02
3. Internship Program Plan	03
4. Technical Aspects	04-13
4.1 Understanding Design theme and making promotional materials.	04-06
4.2 Initial prototype development	07-08
4.3 Final prototype and modification	09-13
5. Technical Result and Outcome	14-16
6. Conclusion	17
7. References	18
8. Plagiarism Report	19

## Table of Figures

<b>S.No</b>	<b>Fig.</b>	<b>Description</b>
01	4.1.1	Promotional material 1
02	4.1.2	Promotional material 2
03	4.1.3	Promotional material 3
04	4.1.4	Promotional material 4
05	4.1.5	Promotional material 5
06	4.1.6	Promotional material 6
07	4.1.7	Initial design for BLOCKGPT(landing page)
08	4.1.8	Initial design for BLOCKGPT(Home page)
09	4.1.9	Simplified and practical version of design
10	4.2.1	Initial integration of platform with BLOCKGPT
11	4.2.2	Web platform implementation based on design(Home Page)
12	4.2.3	Web platform implementation based on design(Landing Page)
13	4.2.4	Modified designing of web based platform
14	4.3.1	Google auth login code snippet
15	4.3.2	Fetching user chat history code snippet
16	4.3.3	Fetching number of sessions code snippet
17	4.3.4	Creating a new chat session code snippet
18	4.3.5	Deleting chat history of session code snippet
19	4.3.6	Sending query code snippet
20	4.3.7	Displaying response by ai code snippet
21	4.3.8	Flow of working of BLOCKGPT
22	4.3.9	Flow of working of client side of BLOCKGPT
23	5.1	Split terminal for opening the BLOCKGPT on local host
24	5.2	The final BLOCKGPT webpage(Landing Page)
25	5.3	BLOCKGPT previous version
26	5.4	The final BLOCKGPT webpage(HomePage)
27	5.5	The final BLOCKGPT webpage(Home Page)

## Introduction

The primary focal point of the startup revolves around delivering financial services. Their principal objective is centered on developing a singular, user-friendly, and secure custodial wallet tailored for cryptocurrency traders. Additionally, they are actively working on offering real-time risk evaluation for both user portfolios and the range of cryptocurrencies available, based on specific parameters.

## Overview

### Description:

Established in 2022, Dolf. Finance is a startup operating within the blockchain domain and was initiated by a trio of individuals - Mr. Kushal Kumar (CEO), Mr. Ravi Chaurasia (CTO), and Mr. Saif Ullah (CMO).

Prior to their venture at Dolf, founders Ravi and Saif had jointly established and managed a blockchain services firm, amassing over a decade of collective experience in blockchain technology and smart contracts. Meanwhile, Mr. Kushal brings a wealth of expertise in the finance sector, having navigated investment banking and risk management intricacies.

Dolf is currently dedicated to the creation of a Decentralized Web3 Bank. Through a proprietary risk assessment framework, they are actively assigning risk ratings to over 200 DeFi protocols and 100+ crypto assets across more than 20 chains, all in "real-time". These ratings serve as a valuable resource for users in identifying optimal DeFi Protocols, while also ensuring the monitoring of portfolio well-being through early alerts. The incorporation of a keyless, self-custody approach guarantees that users have complete control over their funds and eliminates the necessity of safeguarding private keys until access via email, phone, and biometrics is available.

Central to their mission is the aspiration to democratize DeFi, making it accessible to a broader audience.

### Market Opportunity:

As of August 2022, the collective value locked in 127 DeFi protocols reached \$67.87 billion. The global decentralized finance market, valued at \$11.78 billion in 2021, is projected to grow at a CAGR of 42.5% from 2022 to 2030. This growth is propelled by the adoption of DeFi, which has prompted a substantial transformation in the financial sector.

### Fundings:

In Jan 21, 2023, Dolf. Finance raised Pre-seed funding of Rs.12,500,000 from 100x.vc.

### Products:

1. Launched their own crypto trading application on Google Play Store available only outside India.
2. Currently working on the Proprietary Risk Assessment Framework.

### Competitors:

Possible competitors of Dolf. Finance include Jiko, Finja and Picture Wealth.

## Internship Program Plan

During my internship, I was in the SDE branch. Our main goal was to construct language models using machine learning and AI. We also got to work with modern tools like OpenAI and LangChain.

**Start Date:** 29<sup>th</sup> May 2023

**End Date:** 31<sup>st</sup> July 2023

**Mode:** Hybrid

**Duration of Stay:** 3 weeks

Throughout the project timeline, my focus was on creating and constructing the foundational web-based platform for the integration of BlockGPT. I achieved this using the design tools Figma and the development framework React.

My principal task encompassed delving into the startup's brand guidelines to grasp their design patterns, color palettes, and logos. Initially, I concentrated on crafting designs for the startup's promotional materials. Subsequently, I progressed to generating various design iterations and refining them within the Figma platform, ultimately selecting the most suitable design for BlockGPT.

Upon finalizing the chosen design, my efforts shifted to the development phase. I embarked on constructing the frontend of BlockGPT using React and its associated libraries, ensuring a cohesive and user-friendly interface that aligns with the startup's visual identity.

## Technical Aspects

### 4.1 Understanding Design theme and making promotional materials.

Prior to establishing the web-based integration platform for BlockGPT, it was crucial to grasp the design theme and adhere to the brand book guidelines. The promotional materials produced were directly derived from the startup's provided design framework.

In the process of crafting promotional materials, I not only incorporated the provided design aesthetics but also strategically integrated the startup's Unique Selling Proposition (USP) and the benefits offered by Dolf. This approach aimed to effectively communicate the startup's value proposition to the target audience, enhancing engagement and conveying the advantages of the product.

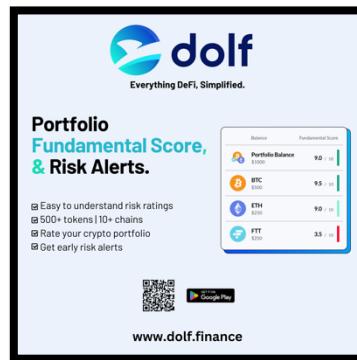


Fig:4.1.1

Fig:4.1.2

Fig:4.1.3



Fig:4.1.4

Fig:4.1.5

Fig:4.1.6

The following phase involved crafting a design for the BLOCKGPT platform. To achieve this, I utilized Figma to create an initial design, drawing inspiration from Dolf's design theme and existing chatbots like ChatGPT. This process aimed to simplify and enhance the platform's appearance, making it both appealing and user-friendly.

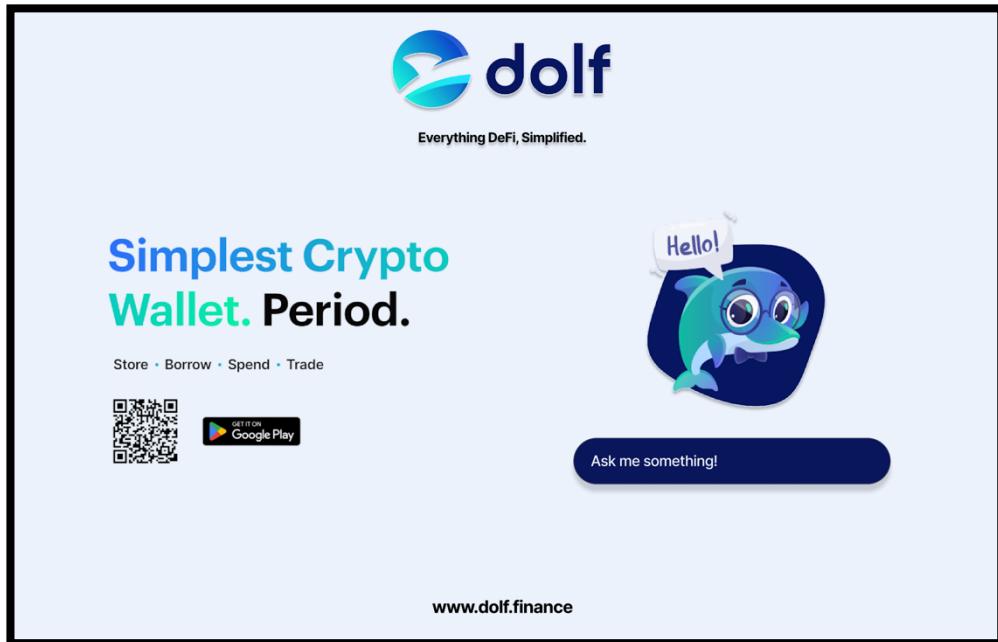


Fig:4.1.7

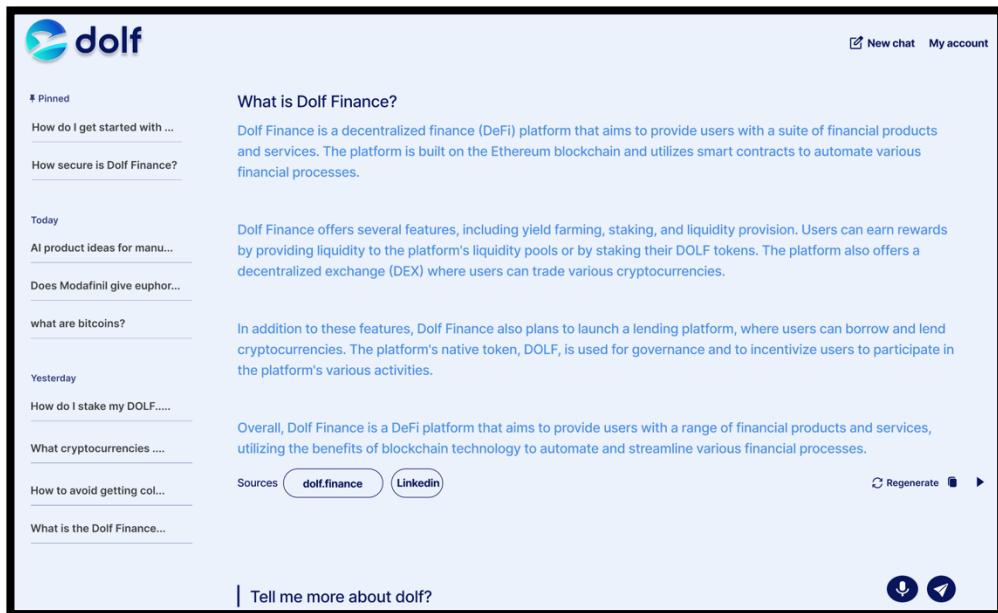


Fig:4.1.8

While the initial concept showed potential, it was unfeasible due to our inability to accommodate all of the elements listed in the design above. Recognizing this limitation, we opted for a much simpler and streamlined user interface (UI) for the web-based platform.

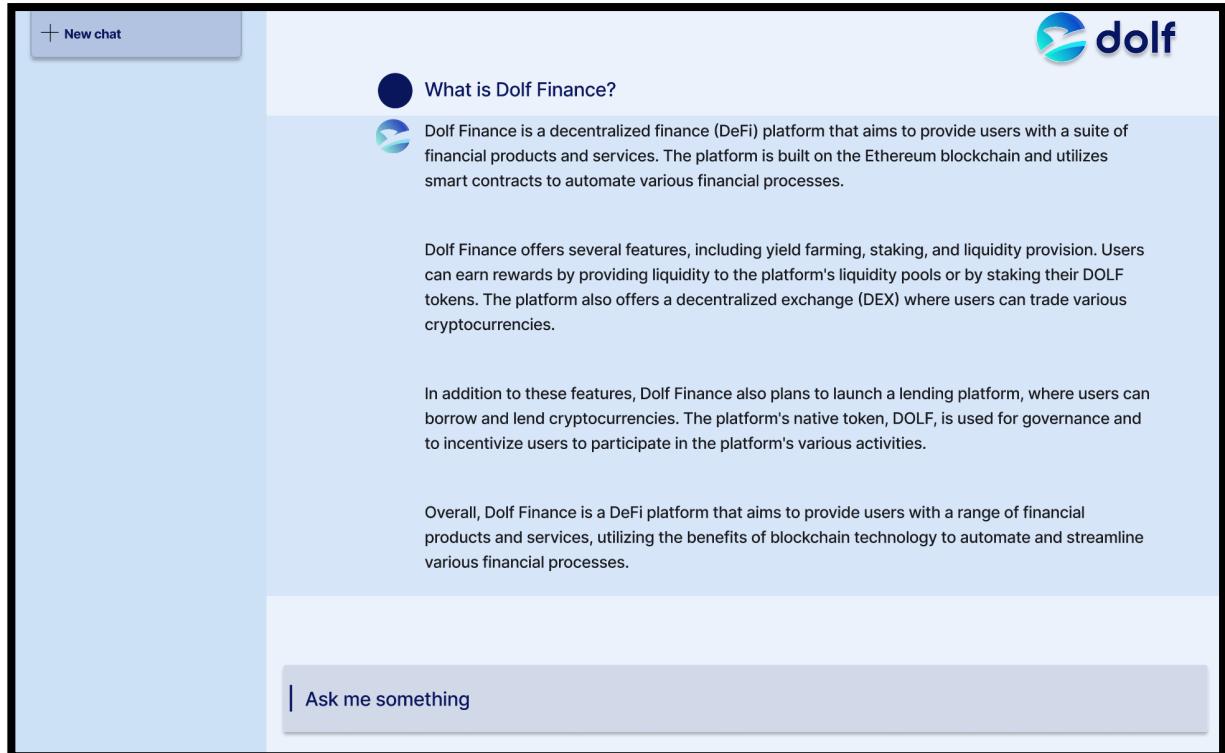


Fig:4.1.9

To sum up the design phase, we started with an exciting design concept inspired by Dolf's theme and other chatbots. However, we quickly realized that some ideas were too complex to put into practice. So, I simplified the design to make it easier for users. This decision set the stage for building the actual platform using React. Our design journey taught us the importance of balancing creativity with practicality, and it got us ready for the next step of turning our ideas into a functional reality.

## 4.2 Initial prototype development

In the initial phase following the design work, I transitioned to developing a basic and functional interface using React. This step aimed to ensure the seamless integration of BLOCKGPT, which was still in its initial stages, using the ChatGPT API key. The primary goal was to create a simple yet practical setup that allowed me to assess the successful integration and interaction of the chatbot within the interface.

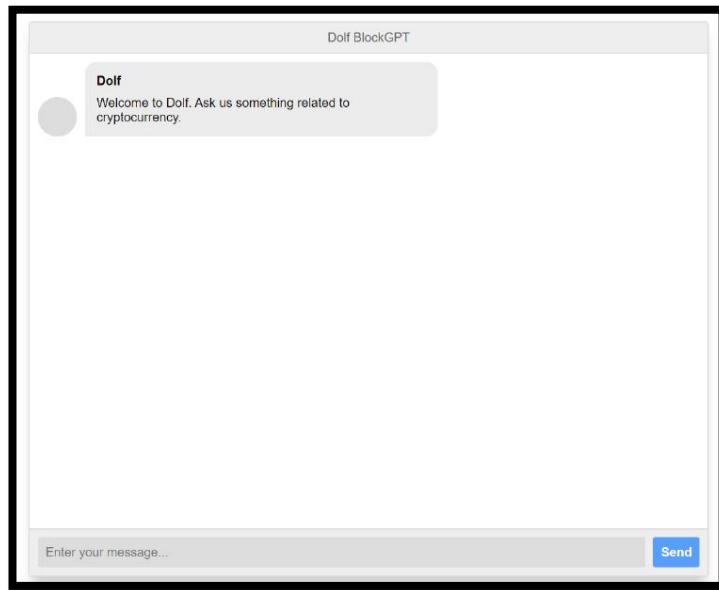


Fig:4.2.1

Following that, I proceeded to enhance the existing usable prototype by developing a more advanced version.

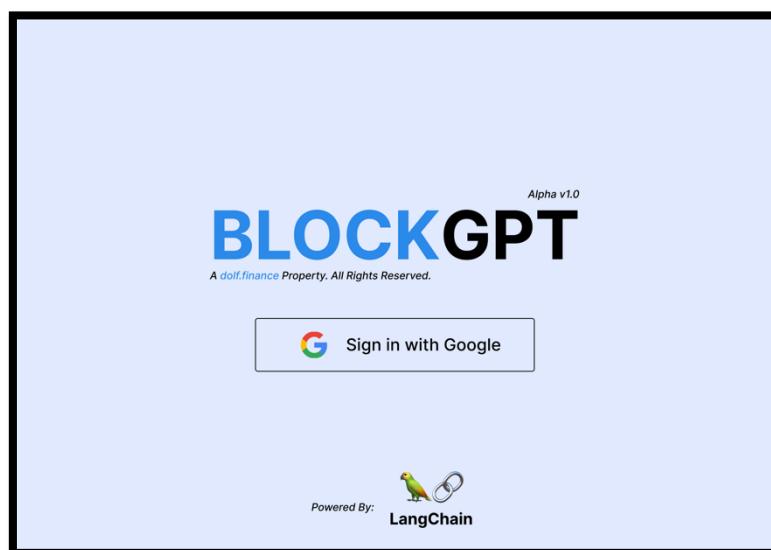


Fig:4.2.2

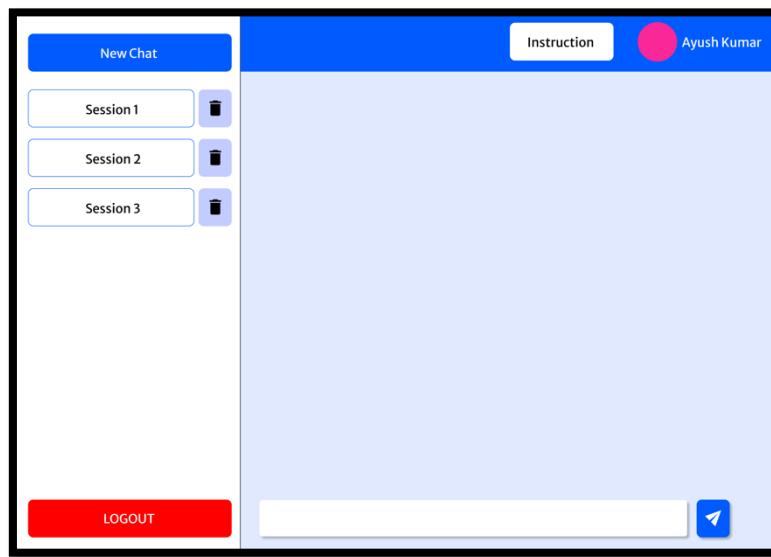


Fig:4.2.3

Subsequently, I implemented modifications to the design to facilitate the seamless integration of the fully developed BLOCKGPT into the interface.

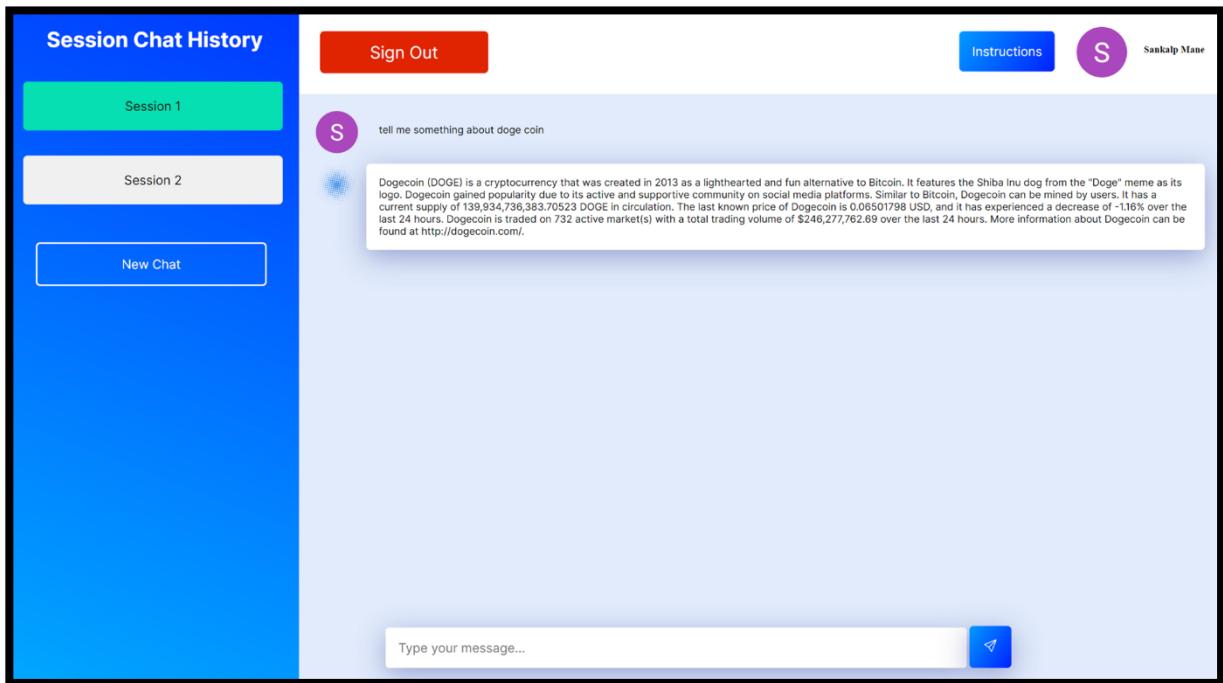


Fig:4.2.4

### 4.3 Final prototype and modification

The final prototype represents the culmination of our work, showcasing a refined and fully functional chatbot interface powered by the advanced BLOCKGPT technology. We made necessary modifications to optimize the user experience and seamlessly integrate the developed BLOCKGPT system, resulting in an intuitive and responsive design. This prototype is a testament to our iterative development approach, effectively combining design enhancements and technological advancements for a powerful and user-friendly blockchain-focused chatbot solution.

1. The given code uses `useEffect` in a React component to:

- 1. Initiate Google Sign-In:** Set up Google Sign-In by initializing with a client ID and callback function.
- 2. Render Button:** Display a Google Sign-In button using the provided HTML element.
- 3. WebSocket Connection:** If logged in, connect via WebSocket, emit user email, and handle real-time updates.
- 4. Update Chat History:** Update chat history when WebSocket receives updates for the selected session.
- 5. Fetch Chat Sessions:** Retrieve chat sessions when logged in.
- 6. Empty Dependency Array:** Ensure the effect runs once on component mount.

In short, the `useEffect` handles Google Sign-In, WebSocket connection, chat history updates, and session fetching on component mount.

```

useEffect(()=>[

  /* global google */
  google.accounts.id.initialize({
    client_id: "215853845060-d4dimifpuqgcerv42nrdblrpdm6vqb9.apps.googleusercontent.com",
    callback: handleCallbackResponse
  });

  google.accounts.id.renderButton(
    document.getElementById("signInDiv"),
    {theme:"outline", size:"large"}
  );

  if (isLoggedIn && user && user.email) {
    // Establish WebSocket connection with the server
    socket.on('connect', () => {
      console.log('WebSocket connected.');
      socket.emit('join', { email: user.email });
    });
  }

  // Handle real-time chat history updates
  socket.on('chat_history_update', (data) => {
    const { session_id, conversation } = data;
    if (session_id === selectedSession) {
      // Update the chat history only if the update is for the selected session
      setChatHistory(conversation);
    }
  });

  // Fetch chat sessions
  fetchChatSessions();
}
]);

```

Fig:4.3.1

2. The given code defines a function called `fetchChatHistory` that fetches chat history for a specific user's selected session. It checks if the user is logged in and a session is selected. If both conditions are met, it sends an HTTP GET request to the server using the `axios` library with the user's email and session as parameters. Upon a successful response, the chat history data is updated using `setChatHistory`. Any errors during the request are logged using `console.error`. This function is crucial for displaying chat history in the application.

```
// Fetching chat history of user email of each selected sessions
const fetchChatHistory = () => {
  if (user && user.email && selectedSession) {
    axios.get(`http://localhost:9000/api/chat-history/${user.email}/${selectedSession}`)
      .then((response) => {
        setChatHistory(response.data);
      })
      .catch((error) => {
        console.error(error);
      });
  }
};
```

Fig:4.3.2

3. The code defines the `fetchChatSessions` function that fetches the user's chat sessions. If the user is logged in and has an email, it sends an HTTP GET request using `axios` to the server with the user's email. On success, it updates chat sessions with `setChatSessions`. If no session is selected yet and there are sessions available, it selects the first session as the default. Errors are logged using `console.error`. This function manages retrieving and displaying chat sessions in the application.

```
// Fetching the number of chat sessions for each user email
const fetchChatSessions = () => {
  if (user && user.email) {
    axios.get(`http://localhost:9000/api/chat-sessions/${user.email}`)
      .then((response) => {
        setChatSessions(response.data);
        if (!selectedSession && response.data.length > 0) {
          setSelectedSession(response.data[0]); // Select the first session by default
        }
      })
      .catch((error) => {
        console.error(error);
      });
  }
};
```

Fig:4.3.3

4. `handleNewChatSession` creates a new chat session by sending a POST request to the server with the user's email. After a successful creation, it updates sessions, selects the new session, clears chat history, and sets a flag. `handleSessionSelect` updates the selected session based on user interaction.

```
// Function if user clicks a new session
const handleNewChatSession = async () => {
  try {
    // Call the backend to start a new chat session
    const response = await axios.post('http://localhost:9000/api/new-chat-session', {
      email: user.email,
    });
    // Refresh the chat sessions list
    fetchChatSessions();
    // Set the newly created session as the selected session
    setSelectedSession(response.data.session_id);
    // Clear chat history when starting a new chat
    setChatHistory([]);
    // Set the flag to indicate a new session has been created
    setNewSessionCreated(true);
  } catch (error) {
    console.error(error);
  }
};

const handleSessionSelect = (session) => {
  setSelectedSession(session);
};
```

Fig:4.3.4

5. `handleDeleteChatHistory` deletes chat history of a specific session. It sends a DELETE request to the server, refreshes chat history, updates sessions, and removes the deleted session from the list. If the deleted session was selected, the selected session state is reset. Errors are handled with logging.

```
//Delete chat history of particular session
const handleDeleteChatHistory = async (session) => {
  try {
    // Delete chat history from MongoDB
    await axios.delete(`http://localhost:9000/api/delete-chat-history/${user.email}/${session}`);
    // After successful deletion, you might want to refresh the chat history
    fetchChatHistory();

    // Remove the deleted session from the list of sessions
    setChatSessions((prevSessions) => prevSessions.filter(s => s !== session));
    // If the deleted session was selected, reset the selectedSession state
    if (selectedSession === session) {
      setSelectedSession(null);
    }
  } catch (error) {
    console.error(error);
  }
};
```

Fig:4.3.5

6. `handleSubmit` handles user input submission. It prevents the default form behavior, initiates a new chat session if none exists, sends user data to the server, displays assistant responses letter by letter, resets flags and data for a clean state, and handles errors.

```
// Handle submit function, sending the user query to the api endpoint
const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    if (!selectedSession) {
      await handleNewChatSession();
    }
    // Sending the required info to the server
    const response = await axios.post('http://localhost:9000/api', {
      message: message,
      email: user.email,
      name: user.name,
      picture: user.picture,
      session_id: selectedSession,
      new_session: newSessionCreated,
    });

    // Displaying the assistant response in real-time letter by letter
    printResponse(response.data.message);
    setMessage('');

    // Reset the flag after sending the message
    setNewSessionCreated(false);
  } catch (error) {
    console.error(error);
  }
}
```

Fig:4.3.6

7. `printResponse` function shows the assistant's reply letter by letter. It initializes a timer, appending characters to the response until the message is fully displayed. The process halts when the complete message is shown, enhancing real-time interaction with users

```
// Displaying the assistant response in real-time letter by letter
const printResponse = (message) => {
  setResponse('');
  let index = -1;
  const timer = setInterval(() => {
    if (index >= message.length) {
      clearInterval(timer);
    } else {
      setResponse(prevResponse => prevResponse + message.charAt(index));
      index++;
    }
  }, 50);
};
```

Fig:4.3.7

Below are the flowcharts diagrammatically explaining the flow of BlockGPT.

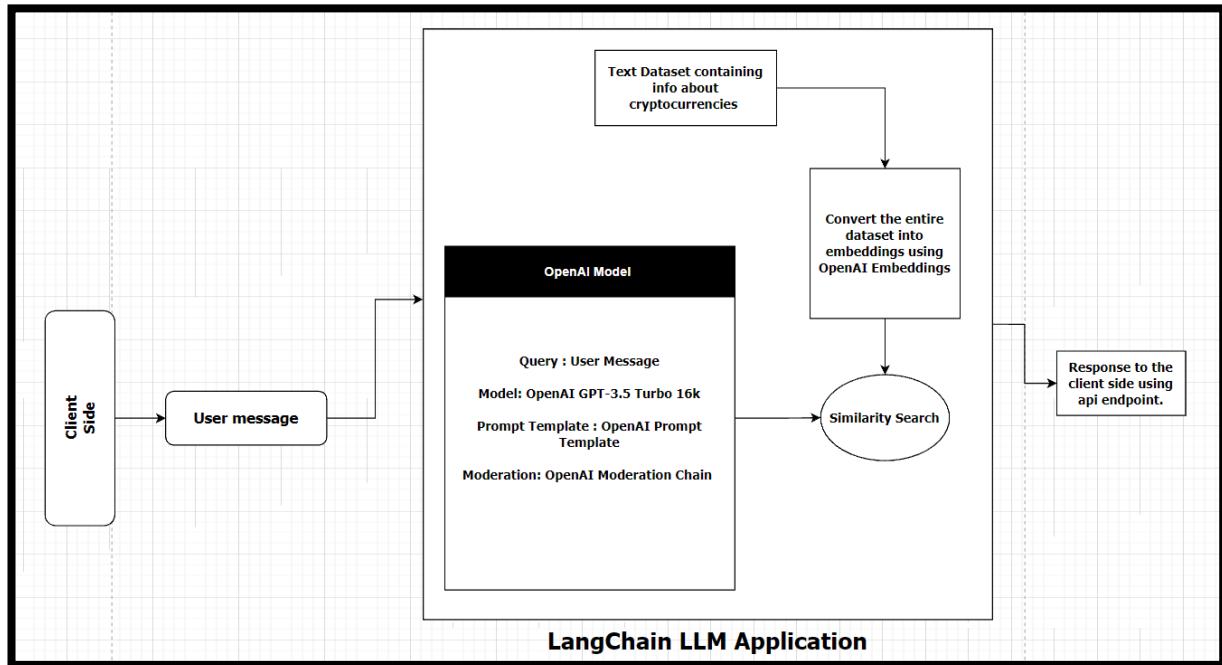


Fig:4.3.8

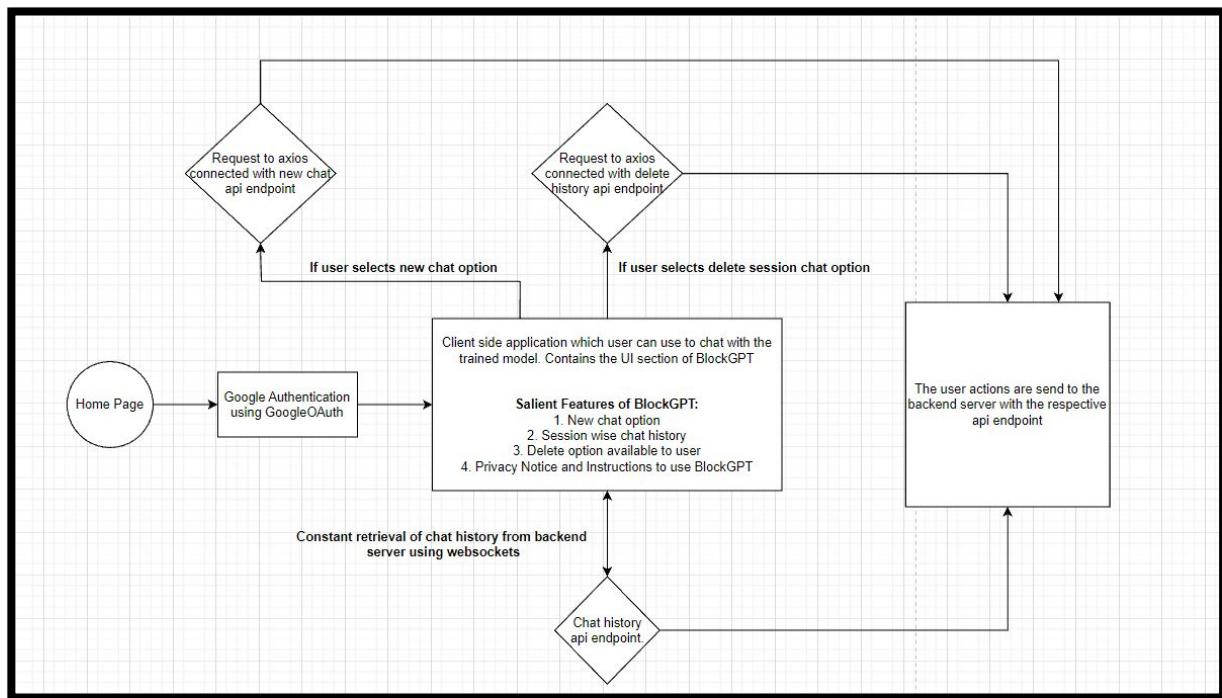
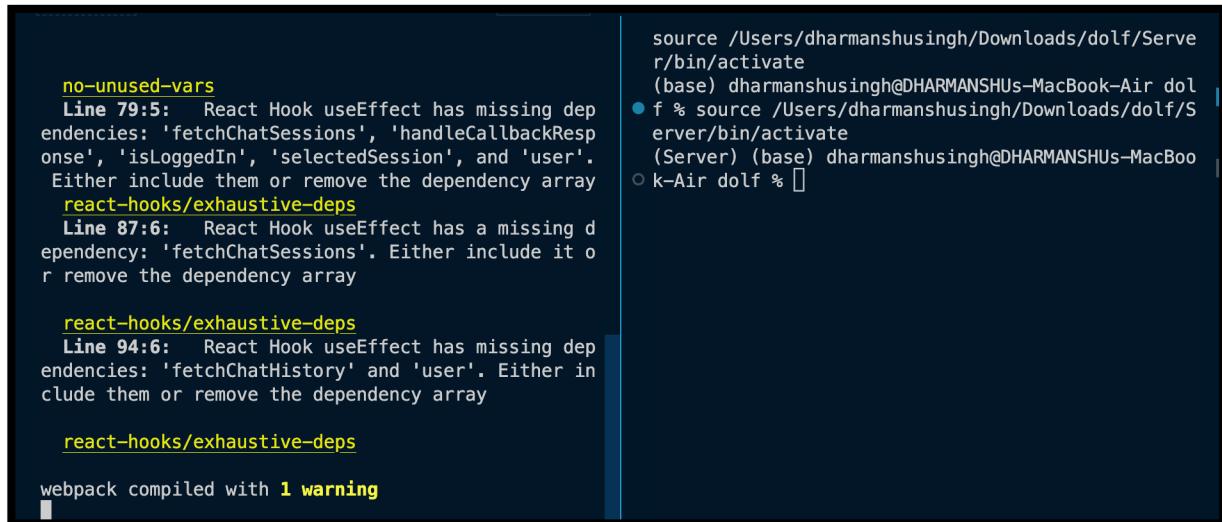


Fig:4.3.9

## Technical Result and Outcome

Presently, BlockGPT operates solely on local systems due to existing bugs in the build and the potential for significant improvements. To initiate BlockGPT on your system, ensure both the client and server are active by utilizing a split terminal in your code editor. Before running the client and server files, it's important to activate the Python environment using the provided Activate.ps1 file.



```

no-unused-vars
Line 79:5:  React Hook useEffect has missing dependencies: 'fetchChatSessions', 'handleCallbackResponse', 'isLoggedIn', 'selectedSession', and 'user'. Either include them or remove the dependency array
react-hooks/exhaustive-deps
Line 87:6:  React Hook useEffect has a missing dependency: 'fetchChatSessions'. Either include it or remove the dependency array

react-hooks/exhaustive-deps
Line 94:6:  React Hook useEffect has missing dependencies: 'fetchChatHistory' and 'user'. Either include them or remove the dependency array

react-hooks/exhaustive-deps
webpack compiled with 1 warning
  
```

Fig:5.1

Below are the screenshots of the α2.0 build.



Fig:5.2

Below image shows BlockGPT's previous version design.

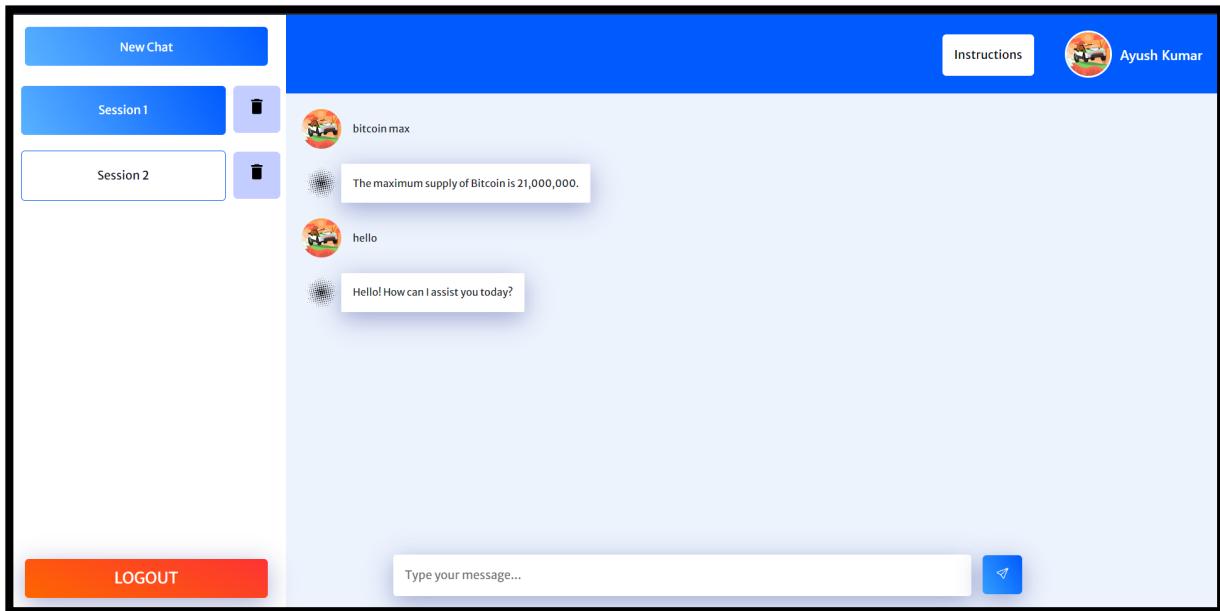


Fig:5.3

Below image shows improved BlockGPT's design

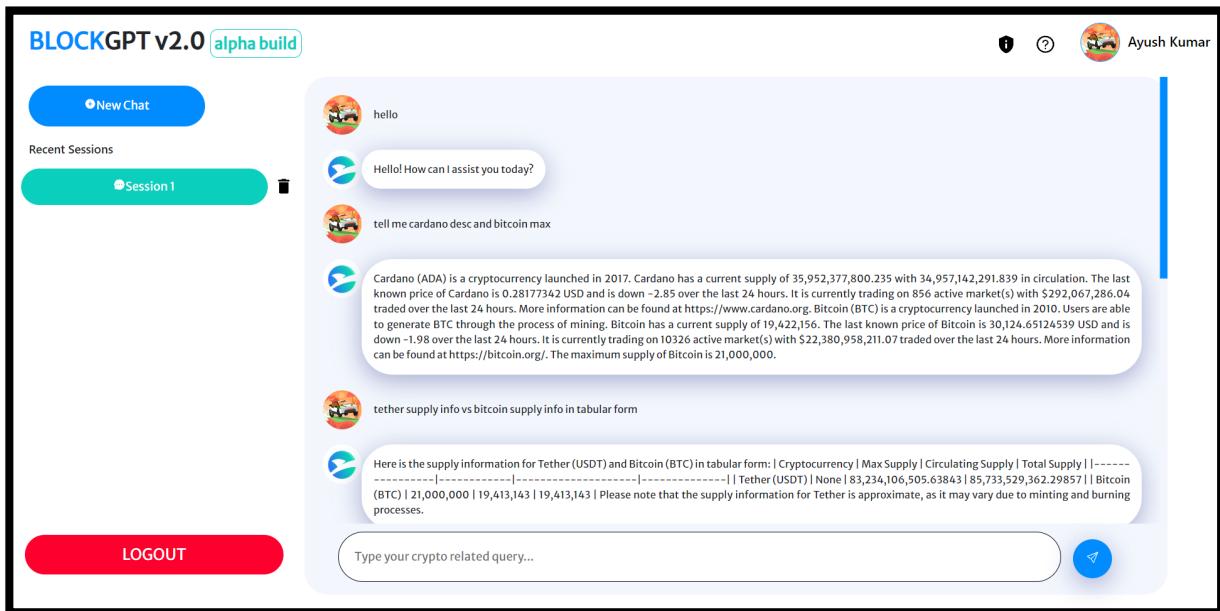


Fig:5.4



Fig:5.5

## Conclusion

17

The alpha build described above contains identified bugs, as detailed in the instruction window. Moving forward, the tasks ahead involve addressing these bugs, enhancing user experience and design, expanding data collection from crypto websites, integrating the final build with the company website, and introducing additional functionalities. These include the implementation of features like storing and displaying chat history along with corresponding timestamps for each user interaction.

## References

18

Andrew Ng, I. F. (2023, 1 1). *Deep Learning - ChatGPT API, Langchain LLM*. Retrieved from Deep Learning: <https://www.deeplearning.ai/short-courses/>

Bruno Krebs, J. C. (2022, 09 15). *Developing RESTful APIs with Python and Flask*. Retrieved from auth0.com: <https://auth0.com/blog/developing-restful-apis-with-python-and-flask/>

Fabisiak, R. (2020, 1 15). *How to create a simple REST API with Python and Flask in 5 minutes*. Retrieved from medium.com: <https://medium.com/duomly-blockchain-online-courses/how-to-create-a-simple-rest-api-with-python-and-flask-in-5-minutes-94bb88f74a23>

Langchain. (2022, 1 1). *Langchain - Documentation*. Retrieved from Langchain: <https://docs.langchain.com/docs/>

OpenAi. (2022, 1 1). *OpenAI - Documentation*. Retrieved from OpenAI: <https://platform.openai.com/docs/introduction>

<https://github.com/anthonyjgrove/react-google-login>

<https://reactjs.org/>

# Plagiarism Report

19

Turnitin Originality Report

Document Viewer

Processed on: 09-Aug-2023 19:41 IST  
ID: 2143538439  
Word Count: 2242  
Submitted: 2

dolf By Dharmanshu Singh

Similarity Index	Similarity by Source
4%	Internet Sources: 1% Publications: 1% Student Papers: 3%

include quoted | include bibliography | excluding matches < 6 words | mode: quickview (classic) report | print | download

1% match (student papers from 24-May-2023)  
[Submitted to South Birmingham College on 2023-05-24](#)

1% match (student papers from 30-May-2023)  
[Submitted to BML Munjal University on 2023-05-30](#)

1% match (student papers from 17-Jun-2023)  
[Submitted to University of Malaya on 2023-06-17](#)

<1% match (Internet from 02-Nov-2022)  
<https://www.gadgetsnow.com/air-purifiers/Electrolux-UltimateHome-500-air-purifier-with-4-stage-filter-for-570sq-ft-room-coverage-Touch-screen-Nordic-Pink-FA41-400PK-price>

<1% match (Internet from 17-Dec-2022)  
<http://www.profindsolutions.com>

<1% match (student papers from 18-Oct-2022)  
[Submitted to Korea University on 2022-10-18](#)

<1% match (student papers from 06-Jul-2023)  
[Submitted to Greenhouse Higher Secondary School on 2023-07-06](#)