CHATBOT FOR HYDROPONICS

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

AMAN CHINMAI DEV BONDLA	21UECB0002	**VTU19464**
SHREYA TIGGA	21UECB0023	**VTU19347**
SOWMIYA E	21UECM0338	**VTU19486**

Under the guidance of
Dr.Arish P,Senior Associate,PwC-PricewaterhouseCoopers
&
Dr. T. KAMALESHWAR, M.Tech., Ph.D.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CHATBOT FOR HYDROPONICS

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

AMAN CHINMAI DEV BONDLA	21UECB0002	**VTU19464**
SHREYA TIGGA	21UECB0023	**VTU19347**
SOWMIYA E	21UECM0338	**VTU19486**

Under the guidance of
Dr.Arish P,Senior Associate,PwC-PricewaterhouseCoopers
&
Dr. T. KAMALESHWAR, M.Tech., Ph.D.,
ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "CHATBOT FOR HYDROPONICS " by AMAN CHINMAI DEV BONDLA (21UECB0002), SHREYA TIGGA (21UECB0023), SOWMIYA E (21UECM0338) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Industry Supervisor
Dr.Arish P
Senior Associate
PwC-PricewaterhouseCoopers
May, 2024

Signature of Supervisor
Dr. T. Kamaleshwar
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Dr. Peter Soosai Anandaraj A
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

AMAN CHINMAI DEV BONDLA

Date: / /

SHREYA TIGGA

Date: / /

SOWMIYA E

Date: / /

APPROVAL SHEET

This project report entitled CHATBOT FOR HYDROPONICS by AMAN CHINMAI DEV BONDLA (21UECB0002), SHREYA TIGGA (21UECB0023), SOWMIYA E (21UECM0338) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr. T. KAMALESHWAR, M.Tech., Ph.D.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Dr. T. KAMALESHWAR, Associate Professor, M.Tech., Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

AMAN CHINMAI DEV BONDLA	**21UECB0002**
SHREYA TIGGA	**21UECB0023**
SOWMIYA E	**21UECM0338**

ABSTRACT

In the realm of modern agriculture, the integration of technology has become increasingly vital, with Natural Language Processing serving as a pivotal tool in bridging the gap between human expertise and computational understanding. This project focuses on the development of an Artificial Intellegence-driven chatbot specifically tailored for hydroponics, a sustainable and innovative method of plant cultivation. The chatbot's primary objective is to provide comprehensive support and guidance to users, staff, and other stakeholders involved in hydroponic farming practices. Leveraging linguistic rules, statistical models, and real-time data integration, the chatbot facilitates tasks such as nutrient solution management, system maintenance, plant care, troubleshooting, and pest control. Furthermore, it offers personalized assistance to users at various stages of their hydroponic journey, from novice enthusiasts to experienced farmers and agricultural professionals. Staff members benefit from the chatbot's training capabilities, which provide continuous education and support in hydroponic techniques and best practices. Client support is also prioritized, ensuring that customers receive timely and accurate assistance with their hydroponic endeavors. Through its seamless integration with multiple platforms and ongoing advancements in deep learning and neural networks, the chatbot aims to revolutionize hydroponic farming practices, fostering sustainability, productivity, and community engagement in the agricultural landscape.

Keywords: Agriculture, Artificial Intelligence, Chatbot, Community Engagement, Customer Service, Nutrient Management.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
CSS	Cascading Style Sheets
DL	Deep Learning
IDE	Integrated Development Environment
JS	Java Script
KB	Knowledge Base
ML	Machine Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
RAM	Random Access Memory
SSD	Solid State Drive
UAT	User Acceptance Testing
UI	User Interface

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Hydroponic farming has emerged as a sustainable and efficient method of plant cultivation, particularly in urban environments and areas with limited arable land, by eliminating the need for soil and utilizing nutrient-rich water solutions. However, successful implementation and management of hydroponic systems require a deep understanding of plant biology, nutrient management, system maintenance, and troubleshooting techniques. To address these challenges, this project focuses on developing an AI-powered chatbot tailored specifically for hydroponics, serving as a virtual assistant to provide users with personalized support and guidance throughout their hydroponic journey. Leveraging natural language processing (NLP), machine learning, and real-time data integration, the chatbot enhances efficiency, productivity, and sustainability in hydroponic farming by offering tailored recommendations on nutrient solution management, plant care techniques, pest control strategies, and system troubleshooting. Additionally, it provides staff training capabilities, ensuring continuous education and support for personnel involved in hydroponic operations. This project aims to revolutionize the way individuals engage with and manage hydroponic systems for sustainable food production, representing a significant advancement in agricultural technology.

1.2 Aim of the project

This project aims to develop an AI-powered chatbot tailored for hydroponic farming, providing personalized support and advice to users while continuously improving its performance through machine learning techniques.

1.3 Project Domain

The project domain revolves around hydroponic farming, a modern agricultural practice that involves cultivating plants without soil. Instead, nutrient-rich water solutions are used to nourish plants, offering advantages such as higher crop yields, reduced water usage, and minimal environmental impact compared to traditional soil-based agriculture. However, successful hydroponic cultivation requires expertise in various areas, including plant biology, nutrient management, system maintenance, and troubleshooting. This project seeks to leverage artificial intelligence (AI) technology to enhance efficiency, productivity, and sustainability in hydroponic farming by developing an AI-powered chatbot tailored specifically for hydroponics.

The AI-powered chatbot serves as a virtual assistant, providing users, staff, and stakeholders with personalized support, guidance, and expertise throughout their hydroponic journey. By utilizing natural language processing (NLP), machine learning, and real-time data integration, the chatbot offers tailored recommendations on nutrient solution management, plant care techniques, pest control strategies, and system troubleshooting. Additionally, it provides staff training capabilities to ensure continuous education and support for personnel involved in hydroponic operations. Overall, this project aims to revolutionize the way individuals engage with and manage hydroponic systems, ultimately contributing to sustainable food production and environmental stewardship.

1.4 Scope of the Project

The scope of this project encompasses the development of an AI-powered chatbot tailored specifically for hydroponic farming. This includes designing and implementing the chatbot's functionality to provide personalized support, guidance, and expertise to users, staff, and stakeholders involved in hydroponic cultivation. The chatbot will utilize advanced technologies such as natural language processing (NLP), machine learning, and real-time data integration to offer tailored recommendations on various aspects of hydroponics, including nutrient solution management, plant care techniques, pest control strategies, and system troubleshooting. Additionally, the chatbot will feature staff training capabilities to ensure continuous education and support for personnel involved in hydroponic operations.

Furthermore, the project will involve the deployment and testing of the chatbot across multiple platforms to ensure accessibility and usability for a wide range of users. User feedback will be collected and analyzed to iteratively improve the chatbot's functionality and user experience. Additionally, the project scope includes ensuring compliance with industry standards for security, privacy, and data protection laws to safeguard user information and maintain trust. Overall, the scope of this project is to develop an innovative and user-friendly solution that enhances efficiency, productivity, and sustainability in hydroponic farming, ultimately contributing to the advancement of agricultural technology and environmental stewardship.

Chapter 2

LITERATURE REVIEW

[1] Hongjiao Liu et. al.,(2022) introduced that recommendation systems are widely used in various industries and are seen as one of the effective methods in reducing information overload. This paper selected three common algorithms for implementation and evaluation among many recommendation algorithms. Two traditional collaborative filtering algorithms - User Collaboration Filter and Item-based Collaborative Filtering; most popular algorithms in the recommendation field - Matrix Factorization.

[2] Jing Yu et. al.,(2021) described that traditional recommendation systems recommend users to personal items according to their pre-behaviors to infer users' personal preference. Although researchers have made some improvements, which take time to factor into traditional recommendation systems, on catching changes in users' interest, it is difficult to always compute the users' preference fluctuations. Therefore, to solve the mentioned above problems, we add personal preference fluctuations into traditional collaborative filtering systems.

[3] Wei Zhao et. al.,(2021) stated that collaborative Filtering is a widely used service recommendation technology. The model calculates the user's recommendation to the service and the user's trust by establishing a recommendation relationship between user services and the trust relationship between users, and by considering the influence of factors such as malicious users on recommendation and trust, increased authenticity and availability of recommendation and trust.

[4] Qingna Pu et. al.,(2021) described that the emerging recommendation systems not only provide users with an excellent user experience and convenience but also help businesses achieve greater profits. This study has chosen a movie recommendation system based on a hybrid recommendation algorithm. This system reduces the time users spend on information searching and enhances their search efficiency, aiming to recommend movies that align with their preferences.

[5] Junhui Huang et. al.,(2021) introduced that in the process of recommendation, finding the similarity between users is the key to the accurate recommendation, so user preference modeling is particularly important. It is of great significance to study the modeling method of network users' preferences and the recommendation algorithm based on the preference model for better-providing information services for users. With the rise of social network research, many researchers combine social networks with personalized recommendations to solve the problems of data sparsity, "cold start", model scalability, and robustness in traditional recommendations, which effectively improves the scalability and accuracy of recommendations.

[6] Han Li et. al.,(2020) proposed that existing music recommendation methods simply use all the music that the user has listened to as the context of music recommendation, resulting in the same context weights learned for different types of music, which seriously affects the accuracy of music recommendation. To improve the accuracy and personalization of music classification, this paper proposes the use of a deep music recommendation algorithm based on user data mining technology in music classification. Tendency, construct behavior feature space, and combine factor decomposition machine theory to predict user behavior types, as the basis for personalized music recommendation.

[7] Wendan Jiang et. al.,(2020) introduced that users need to accurately and quickly obtain the marketing content they need through the media's personalized recommendation system. This paper designs a media-personalized recommendation system based on a network algorithm, discusses the functional requirements of media-personalized recommendation, and the classification of a personalized recommendation system based on a network algorithm. Through the experimental test of the media personalized recommendation system, taking Weibo as an example, the data ratio between the number of effective recommendations and several users in each category is tested.

[8] Zhijie Lan et. al.,(2019) proposed that the traditional graduate employment recommendation system mainly relies on the employment information provided by graduates. The article designs an intelligent recommendation system for graduate employment based on artificial intelligence. Based on the collection and processing of graduates' job-seeking information, the system provides graduates with multi-

dimensional and personalized employment information recommendation services through machine learning technology and recommendation algorithms. It improves the success rate of job hunting for graduates and realizes precise and intelligent employment recommendations for college students.

[9] Dajun Zeng et. al.,(2019) described that music playback platforms would adopt a personalized recommendation method to push music that meets users' tastes to improve their experience and stickiness, so the issue of whether the personalized recommendation is accurate becomes the key. The Look-alike algorithm can identify people with potential associations, while the K-means algorithm can classify them based on the potential connections of data, making them fully feasible for application in music recommendation systems. This article designed a music recommendation system by studying the Look-alike and K-means algorithms and conducted experiments on the designed system. Through experiments, it was proved that the system designed in this article could effectively improve the accuracy of personalized recommendations.

[10] Xian Chen et. al.,(2019) explained that with the improvement of people's living standards, people's spiritual needs continue to increase, which makes personalized travel recommendations more and more popular. However, the complicated and overloaded information on the internet has caused trouble for people to choose travel scenic spots. Although collaborative filtering is widely used for recommendation algorithms, neither user-based nor item-based collaborative filtering takes the users' personalized characteristics into account. To recommend better travel scenic spots to users, we propose a personalized collaborative fusion algorithm combining a user-based original collaborative filtering matrix with users' personalized preference matrix to make the recommendation more accurate and attractive.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing hydroponics chatbot systems are built upon traditional recommendation algorithms, such as collaborative filtering and matrix factorization, aiming to provide tailored guidance and support to users engaged in hydroponic farming. Despite their widespread use, these systems encounter certain limitations that impede their effectiveness. One major challenge is the difficulty in adapting to evolving user preferences and behavior. As users' interests fluctuate over time, the chatbot may struggle to accurately capture these changes, resulting in recommendations that are no longer relevant or useful to the user. This can lead to a decrease in user engagement and satisfaction, undermining the overall utility of the chatbot in assisting hydroponic farmers.

Moreover, traditional hydroponics chatbot systems often face issues related to data sparsity and scalability. In the context of hydroponic farming, where user interaction data may be limited, these systems may struggle to generate accurate recommendations. The lack of sufficient data can compromise the quality of recommendations, leading to suggestions that fail to meet users' needs or expectations. Additionally, as the user base and data volume grow, traditional recommendation algorithms may encounter challenges in efficiently processing and analyzing large datasets, resulting in performance bottlenecks and reduced responsiveness. Addressing these limitations is essential to improve the relevance and reliability of recommendations provided by hydroponics chatbot systems.

3.2 Proposed System

The proposed hydroponics chatbot aims to cater to a diverse range of users, including both experienced hydroponic farmers and individuals new to the practice, as well as staff members involved in hydroponic operations. By targeting this broad audience, the chatbot seeks to provide valuable support and guidance to users at various stages of their hydroponic journey. For experienced farmers, the chatbot will offer advanced tips and techniques to optimize their hydroponic systems and maximize crop yields. Meanwhile, individuals new to hydroponics will receive beginner-friendly advice and step-by-step instructions to help them get started with hydroponic farming. Additionally, staff members involved in hydroponic operations, such as technicians and customer support representatives, will benefit from the chatbot's training capabilities, which will provide them with continuous education and support in hydroponic techniques and best practices.

By catering to users, non-users, and staff members, the hydroponics chatbot aims to create a comprehensive support system for hydroponic farming. Users will have access to personalized recommendations and troubleshooting assistance, while non-users will receive informative content and resources to help them explore the benefits of hydroponics and make informed decisions. Staff members, on the other hand, will benefit from training modules and real-time support, enabling them to better assist users and address any issues that may arise. Overall, the proposed chatbot seeks to democratize access to hydroponic farming knowledge and expertise, empowering individuals from all backgrounds to engage in sustainable and efficient agriculture practices.

3.3 Feasibility Study

3.3.1 Economic Feasibility

The economic feasibility of deploying a hydroponics chatbot hinges on its ability to streamline operations, reduce costs, and generate revenue. By automating routine tasks and inquiries, the chatbot frees up human resources and enhances operational efficiency. This translates into cost savings for the business. Additionally, the chatbot can serve as a revenue-generating tool through subscription models, premium features, and promoting hydroponic products and services. By analyzing the initial investment required for development against the projected returns, businesses can assess the chatbot's ROI and determine its economic viability. Overall, the hydroponics chatbot presents an opportunity for businesses to optimize resources, drive revenue, and achieve economic feasibility in the rapidly evolving landscape of hydroponic farming.

3.3.2 Technical Feasibility

The technical feasibility of implementing a hydroponics chatbot involves assessing whether the necessary technology and infrastructure are available to support its development and operation. Firstly, it requires robust natural language processing (NLP) algorithms to accurately understand and respond to user queries in a conversational manner. Additionally, the chatbot must be integrated with hydroponics-specific knowledge databases and real-time data sources to provide relevant and up-to-date information to users. Furthermore, ensuring compatibility with various platforms and devices is essential to reach a wide audience of users. From a development standpoint, the availability of skilled programmers and developers proficient in AI, NLP, and chatbot development tools is crucial. Moreover, the scalability and reliability of the infrastructure supporting the chatbot, including servers and databases, must be considered to accommodate potential growth in user traffic and data volume. Overall, by evaluating these technical aspects, businesses can determine the feasibility of implementing a hydroponics chatbot and ensure its successful development and deployment.

3.3.3 Social Feasibility

The social feasibility of implementing a hydroponics chatbot involves evaluating its acceptance and impact on various stakeholders, including users, staff members, and the wider community. Firstly, it's essential to assess the level of acceptance and willingness of users to interact with a chatbot for hydroponic farming guidance and support. Conducting user surveys or focus groups can provide valuable insights into user preferences and expectations regarding the chatbot's functionality and features. Additionally, engaging with staff members and addressing any concerns or resistance to change is crucial for successful implementation. Furthermore, considering the broader societal impact of the chatbot, such as its contribution to promoting sustainable agriculture practices and enhancing food security, is essential. By fostering awareness and education about hydroponic farming through the chatbot, businesses can build trust and credibility within the community. Overall, by addressing social concerns and ensuring alignment with user needs and societal values, the hydroponics chatbot can achieve social feasibility and positively impact the community.

3.4 System Specification

3.4.1 Hardware Specification

1. Processor: Intel Core i9-11900K (11th Gen), 3.5 GHz (up to 5.3 GHz with Turbo Boost), 8 cores
2. RAM: 32 GB DDR4 (4000 MHz)
3. Storage: 1 TB PCIe NVMe SSD
4. Graphics: NVIDIA GeForce RTX 3080, 16 GB GDDR6
5. Display: 17.3-inch QHD (2560 x 1440) IPS display with 165Hz refresh rate
6. Connectivity: Wi-Fi 6E (802.11ax), Bluetooth 5.2
7. Ports:
- 3 x USB 3.2 Gen 2
- 1 x USB-C 4.0
- HDMI 2.1
- RJ45 Ethernet
- Audio Jack
8. Operating System: Windows 11 Pro
9. Battery: 8-cell 99Whr Li-polymer battery
10. Dimensions: 15.7 x 11.8 x 1.1 inches
11. Weight: 6.5 lbs

3.4.2 Software Specification

- Operating System: Windows 11 Pro
- Programming Language: Python 3.9
- Chatbot Framework: Rasa Open Source 3.0
- Natural Language Processing (NLP) Library: spaCy 3.0
- Machine Learning Framework: TensorFlow 2.7

- Database Management System: PostgreSQL 14
- Web Development Framework: Flask 2.0
- Version Control: Git 2.35
- Integrated Development Environment (IDE): Visual Studio Code 1.65
- Virtual Environment Management: Anaconda 2022.02

3.4.3 Standards and Policies

Anaconda Prompt:

The Anaconda Prompt serves as a command-line interface specifically designed for managing Machine Learning (ML) modules within the hydroponics chatbot project. It provides a convenient environment for handling ML libraries and dependencies, facilitating efficient development and deployment processes. Anaconda Navigator, available across Windows, Linux, and MacOS platforms, offers a range of Integrated Development Environments (IDEs) that streamline coding tasks. The versatility of Anaconda allows seamless integration with various ML frameworks, making it an ideal choice for developing the chatbot's backend functionalities.

Standard Used: ISO/IEC 27001

Jupyter: Jupyter plays a crucial role in the hydroponics chatbot project by providing an interactive environment for creating and sharing documents containing live code, visualizations, and narrative text. It serves as a powerful tool for data analysis, model training, and experimentation, enabling effective collaboration among project team members. By adhering to ISO/IEC 27001 standards, the project ensures the security and integrity of data processed within the Jupyter environment, safeguarding sensitive information and maintaining compliance with industry regulations.

Standard Used: ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Chatbot's Architecture

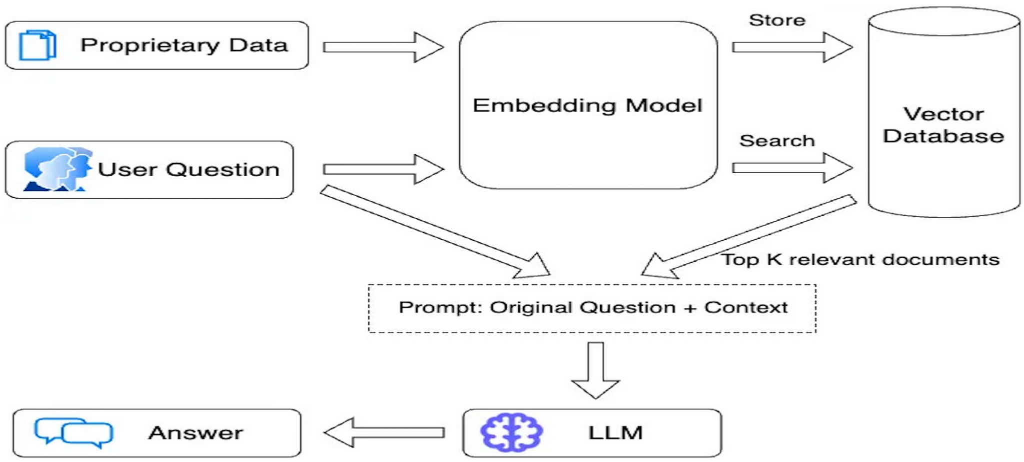

Figure 4.1: **Chatbot Architecture**

In figure 4.1, it showcases the system design of the hydroponics chatbot, highlighting its core components and interactions. At the heart of the diagram lies the Chatbot Engine, responsible for processing user inputs and generating responses. Surrounding the engine are key components such as the User Interface (UI) for user interaction, the Natural Language Understanding (NLU) module for interpreting user messages, and the Knowledge Base (KB) containing information on hydroponic farming. Backend services handle tasks like user authentication and data storage, while analytics tools track user interactions and feedback for continuous improvement. Overall, the architecture demonstrates how the chatbot seamlessly integrates various technologies to provide users with accurate and helpful information on hydroponic farming.

4.2 Design Phase

4.2.1 Data Flow Diagram

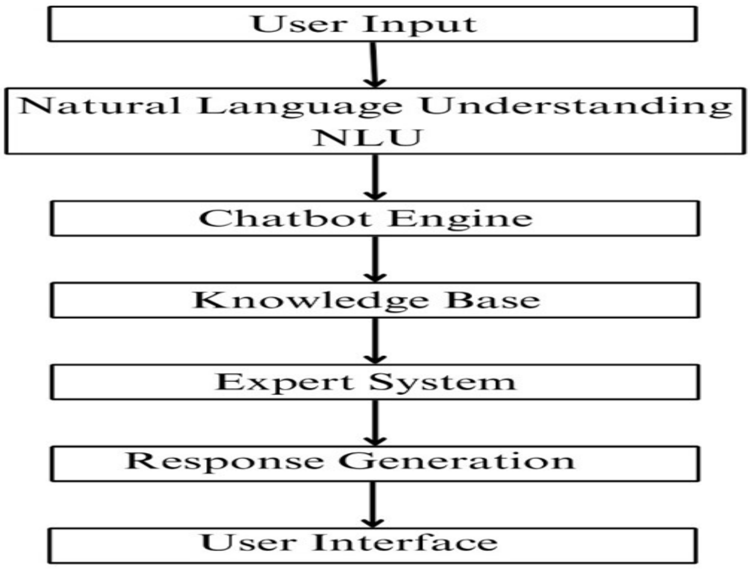

Figure 4.2: **Data Flow for Chatbot**

In figure 4.2, it visualize the flow of data within a system. It illustrates how information moves between different processes, data stores, and external entities. In the context of the hydroponics chatbot project, a DFD would depict the flow of information between components such as the user interface, the chatbot engine, backend services, and external data sources. It would show how user inputs are received, processed, and transformed into meaningful responses by the chatbot engine. Additionally, the DFD would illustrate how data is stored and retrieved from the knowledge base and other data repositories. Overall, the DFD provides a comprehensive overview of how data moves through the system, helping to identify potential bottlenecks, inefficiencies, and opportunities for optimization.

4.2.2 Use Case Diagram

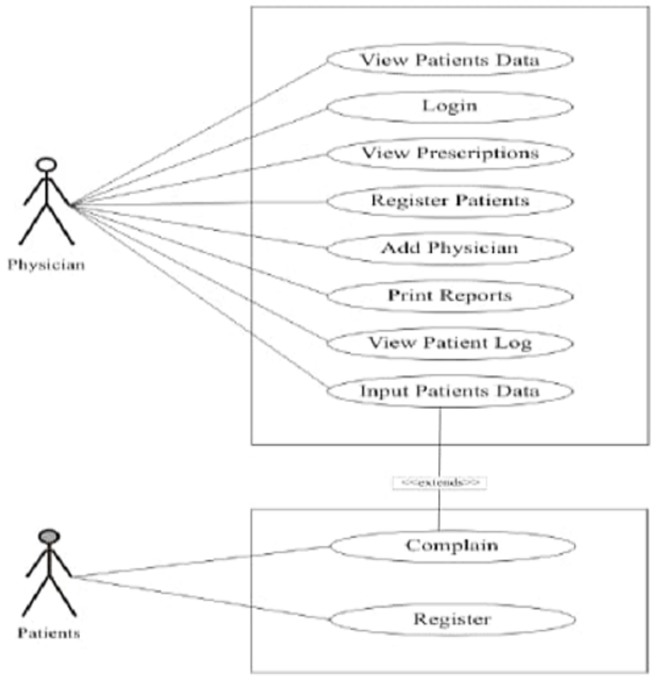

Figure 4.3: **Use Case Diagram of Chatbot**

In figure 4.3, it illustrates the various interactions between users (actors) and the system, highlighting the different functionalities or tasks that users can perform. In the context of the hydroponics chatbot project, a use case diagram would identify the different actors involved, such as hydroponic farmers, novice users, staff members, and administrators. Each actor would be associated with specific use cases representing the actions they can take within the system.

4.2.3 Class Diagram

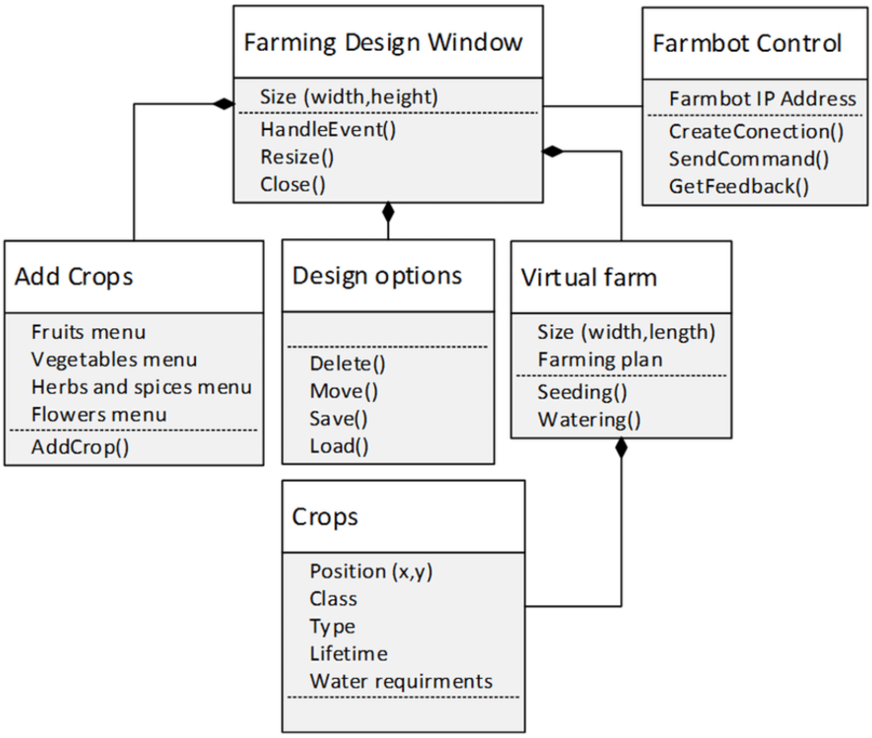

Figure 4.4: **Class Diagram for Chatbot**

In figure 4.4, it shows that the hydroponics chatbot project would outline the structure of the system by defining classes, attributes, methods, and relationships between them. Key classes include Chatbot for core functionality, Users for different user roles, and components like Natural Language Understanding (NLU) and Knowledge Base (KB). These classes encapsulate the system's functionalities, such as processing user inputs and managing conversation flows. Relationships between classes, depicted through associations, illustrate how components interact within the system. For instance, the Chatbot class may have a composition relationship with NLU to indicate its dependency on NLU functionality. Overall, the class diagram provides a comprehensive overview of the system's architecture, aiding in understanding its design and implementation.

4.2.4 Sequence Diagram

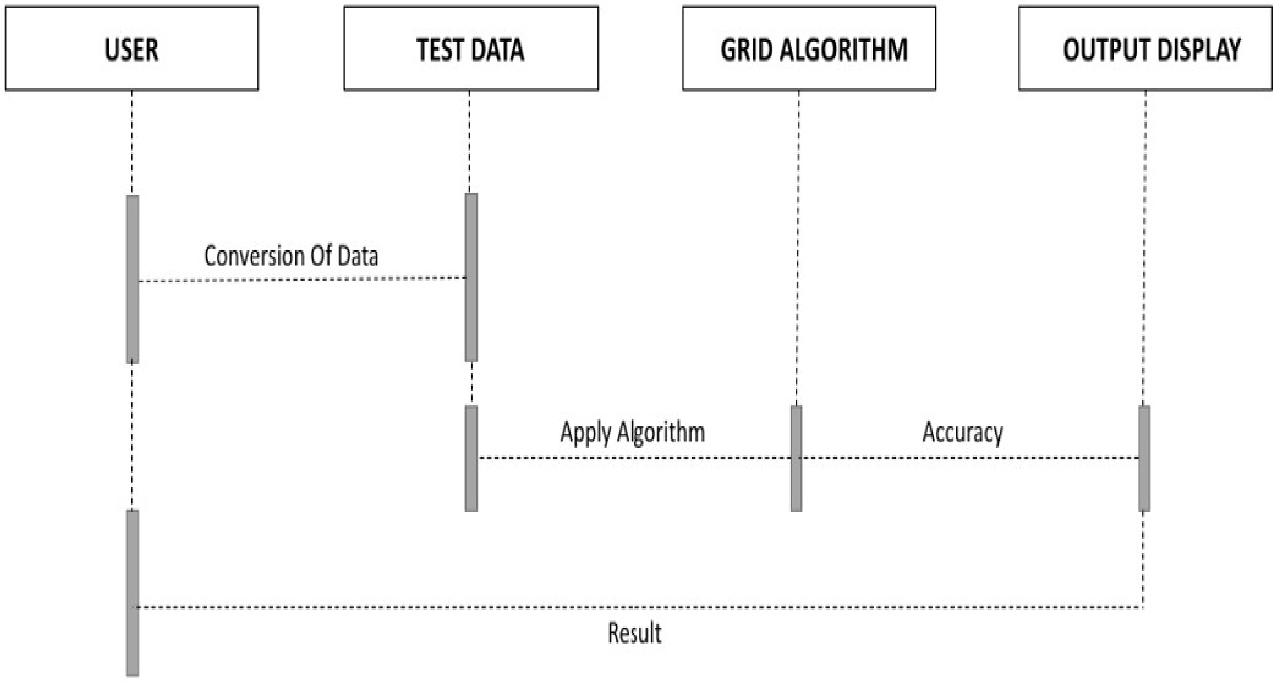

Figure 4.5: **Sequence Diagram for Chatbot**

In figure 4.5, it shows that the hydroponics chatbot project would depict the chronological flow of interactions between various components or objects within the system. It would illustrate how messages are exchanged between different entities over time to achieve specific functionalities. For instance, the diagram might show the sequence of steps involved in a user querying the chatbot, the chatbot processing the query, retrieving relevant information from the knowledge base, and generating a response back to the user. Each step in the sequence would be represented by a lifeline for the corresponding object, with arrows indicating the flow of messages between them.

4.2.5 Activity Diagram

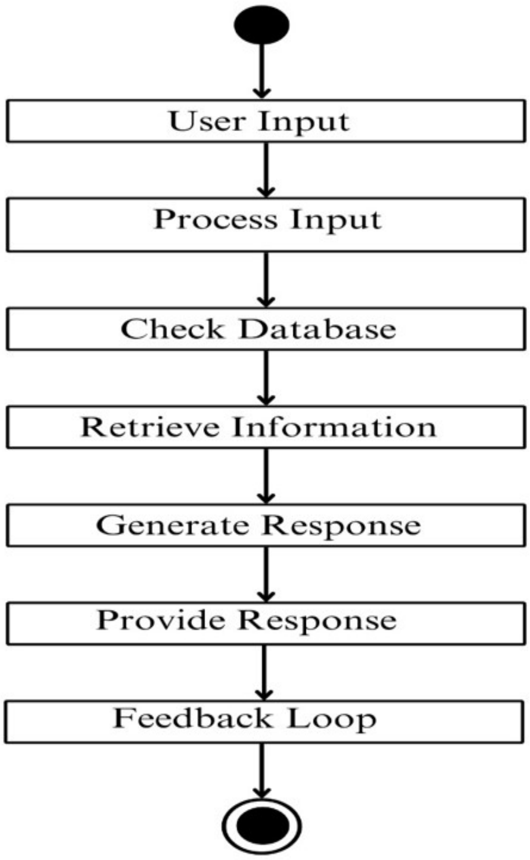

Figure 4.6: **Activity Diagram for Chatbot**

In figure 4.6, it provides a visual representation of the flow of activities or actions within a system, illustrating the sequence of steps involved in completing a specific process or use case. In the context of the hydroponics chatbot project, an activity diagram would outline the steps taken to perform various tasks related to hydroponic farming, such as getting recommendations for crop cultivation or troubleshooting common issues. Each activity would be represented by a rounded rectangle, and the flow between activities would be indicated by arrows. Decision points or branches in the process would be depicted using diamond-shaped nodes, allowing for different paths or outcomes based on certain conditions.

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

For the hydroponics chatbot project, algorithms play a pivotal role in enhancing user interaction and providing valuable assistance. The Natural Language Understanding (NLU) algorithm is essential for accurately interpreting user queries and extracting relevant information. By leveraging techniques like deep learning models, such as BERT or Transformer, the chatbot can understand user intents and identify key entities related to hydroponic farming tasks or inquiries. Additionally, Recommendation Algorithms are employed to offer personalized suggestions for tasks like crop cultivation and troubleshooting. Collaborative filtering or hybrid approaches enable the chatbot to generate tailored recommendations based on user preferences and historical interactions, enhancing the user experience and guiding users towards optimal solutions.

Furthermore, Machine Learning (ML) algorithms are employed for various tasks within the chatbot, ranging from sentiment analysis to predict user satisfaction to classification and regression algorithms for data analysis and prediction. These algorithms enable the chatbot to categorize user queries, predict trends in hydroponic farming, and continuously optimize its responses based on user feedback. Overall, the integration of these algorithms empowers the hydroponics chatbot to deliver personalized, accurate, and timely assistance, fostering a more efficient and user-friendly experience for hydroponic farmers and enthusiasts alike.

4.3.2 Pseudo Code

```
Initialize Chatbot

while True:
    user_input = get_user_input()

    if user_input contains "hello" or "hi":
        respond("Hello! How can I assist you with hydroponics today?")

    else if user_input contains "hydroponics" or "hydroponic system":
        respond("Hydroponics is a method of growing plants without soil, using nutrient-rich water
            instead. What specific information are you looking for?")

    else if user_input contains "types of hydroponic systems":
```

```
        respond("There are several types of hydroponic systems, including Deep Water Culture,
            Nutrient Film Technique, Ebb and Flow, Drip System, and Aeroponics. Would you like more
            details on any of these?")

    else if user_input contains "nutrients for hydroponics":
        respond("Nutrients for hydroponics typically include essential macro and micronutrients
            necessary for plant growth, such as nitrogen, phosphorus, potassium, calcium, magnesium,
            and others. Would you like recommendations on specific nutrient solutions?")

    else if user_input contains "pH level":
        respond("Maintaining the proper pH level is crucial in hydroponics. Most plants prefer a pH
            range between 5.5 and 6.5. Have you tested the pH level of your nutrient solution
            recently?")

    else if user_input contains "lighting for hydroponics":
        respond("Lighting is essential for plant growth in hydroponics. Full-spectrum LED grow
            lights are commonly used because they provide the necessary spectrum for photosynthesis.
            Would you like recommendations on lighting setups?")

    else if user_input contains "common hydroponic pests":
        respond("Common pests in hydroponic systems include aphids, spider mites, whiteflies, and
            fungus gnats. Preventive measures like maintaining cleanliness and using beneficial
            insects can help control them. Do you need advice on pest management?")

    else if user_input contains "how to start a hydroponic garden":
        respond("Starting a hydroponic garden involves setting up a suitable system, selecting the
            right plants, ensuring proper nutrient balance, and maintaining environmental conditions
            like temperature and humidity. Would you like a step-by-step guide?")

    else if user_input contains "thank you" or "thanks":
        respond("You're welcome! If you have any more questions, feel free to ask.")

    else:
        respond("I'm sorry, I didn't understand that. Can you please rephrase or ask another
            question?")
```

4.4 Module Description

4.4.1 User Interface (UI) Module

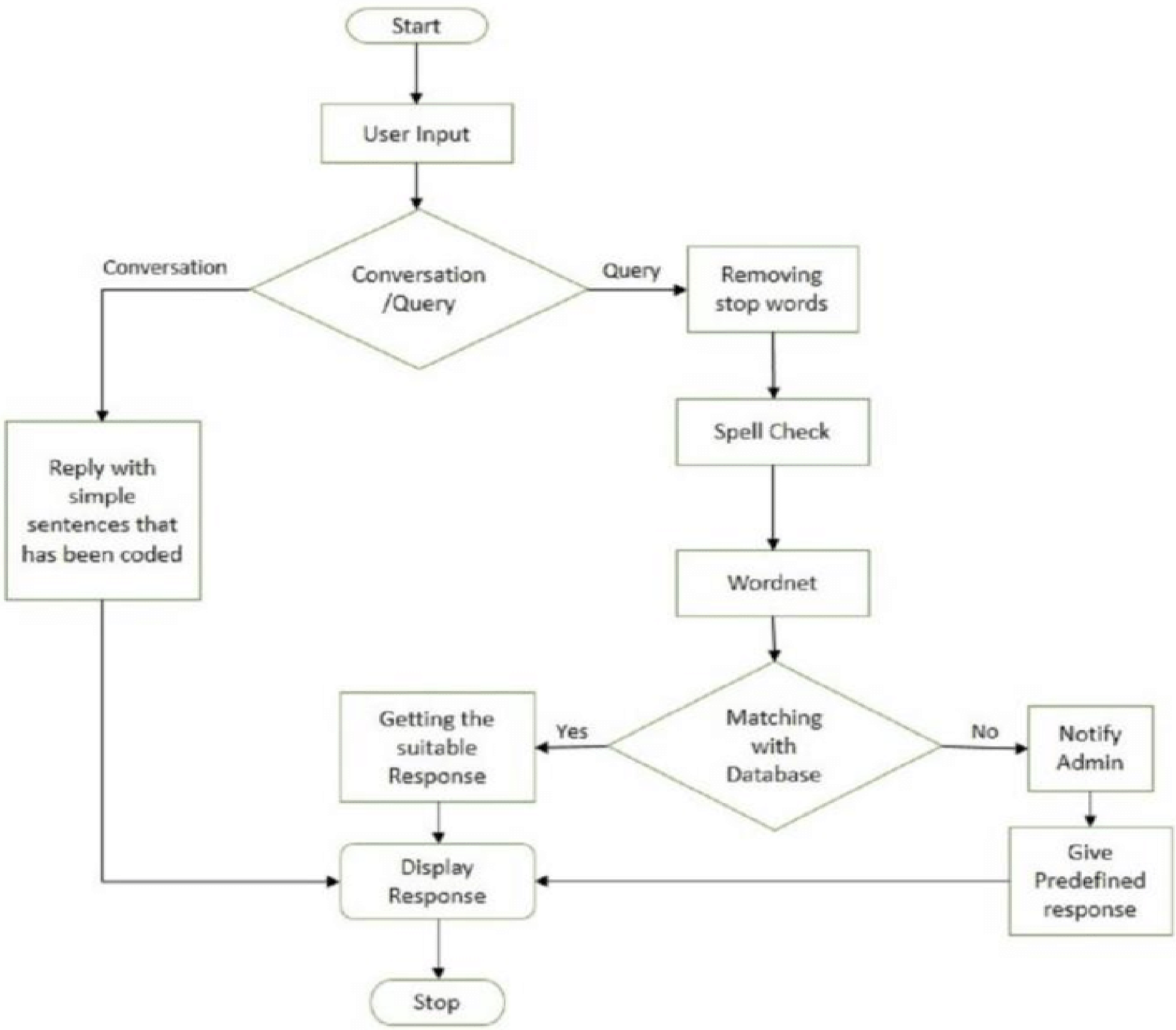

Figure 4.7: **User Interface Module Diagram**

The User Interface (UI) module is responsible for providing an intuitive and user-friendly interface for interacting with the hydroponics chatbot. It encompasses the graphical elements and functionalities that enable users to input queries, receive responses, and navigate through the chatbot's features. The UI module may include components such as text input fields, buttons, dropdown menus, and displays for showing chatbot responses in a visually appealing manner. It ensures a seamless and engaging user experience by presenting information in a clear and organized format, facilitating easy access to hydroponic farming advice and assistance.

4.4.2 Chatbot Engine Module

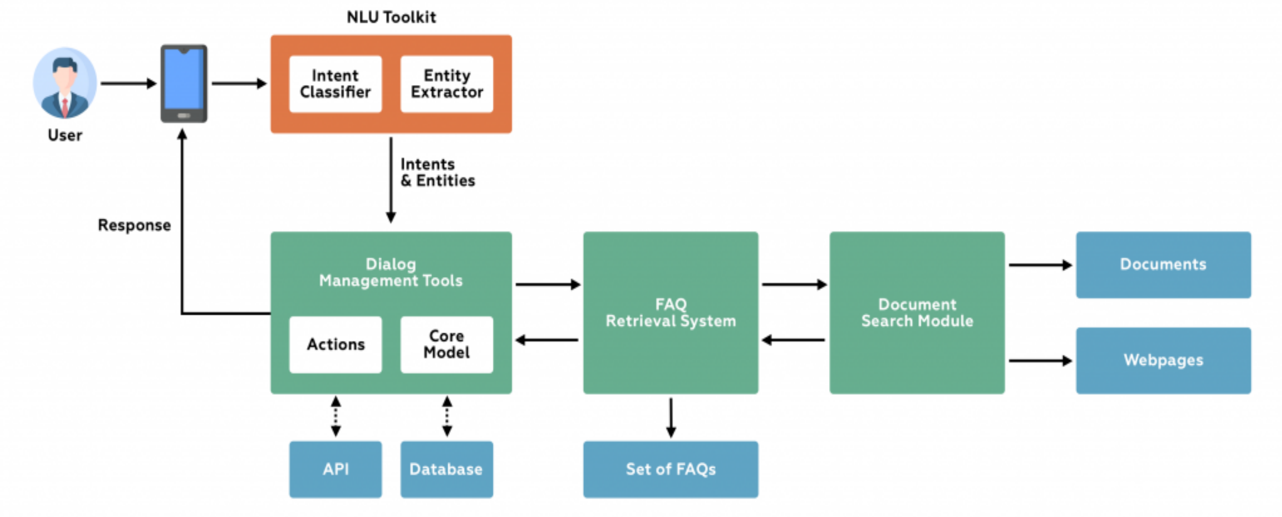

Figure 4.8: **Chatbot Engine Module Diagram**

The Chatbot Engine module serves as the central component of the hydroponics chatbot, responsible for processing user inputs, generating responses, and managing conversation flows. It incorporates sophisticated algorithms and natural language processing (NLP) techniques to understand user queries, extract relevant information, and formulate appropriate responses. The Chatbot Engine utilizes Natural Language Understanding (NLU) algorithms to interpret the intent and context of user messages, enabling it to provide accurate and contextually relevant answers. Additionally, the module may leverage machine learning models to continuously learn from user interactions, improve its understanding, and enhance the quality of responses over time. It acts as the intelligent backbone of the chatbot, enabling it to effectively communicate with users and offer personalized assistance tailored to their needs.

4.4.3 Knowledge Base (KB) Module

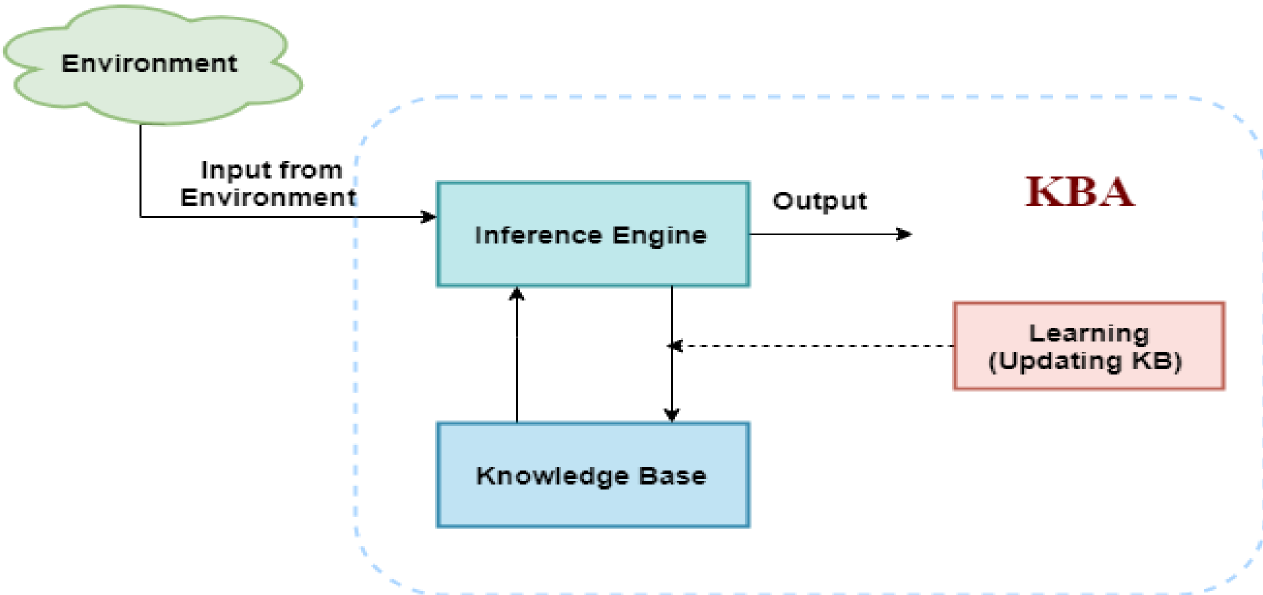

Figure 4.9: **Knowledge Base Diagram for Chatbot**

The Knowledge Base (KB) module serves as the repository of information related to hydroponic farming, providing a comprehensive database of resources, guidelines, and recommendations. It encompasses a wide range of topics, including crop cultivation techniques, nutrient management, pest control, troubleshooting guides, and product recommendations. The KB module stores structured data, textual information, and multimedia content, organized in a hierarchical manner for easy retrieval and access. It acts as the knowledge backbone of the chatbot, enabling it to access relevant information and provide accurate responses to user queries. The KB module may integrate with external databases, APIs, and online resources to ensure that the chatbot has access to the latest and most reliable information in the field of hydroponics. Additionally, it may incorporate mechanisms for content updating, version control, and quality assurance to maintain the accuracy and relevance of the knowledge base over time.

4.5 Steps to execute/run/implement the project

4.5.1 Setting Up Development Environment

- Install Python: Download and install the latest version of Python from the official website (https://www.python.org/https://www.python.org/). Ensure that

Python is added to the system PATH during installation.

- Install Required Libraries: Use pip, the Python package manager, to install necessary libraries such as NLTK (Natural Language Toolkit) and TensorFlow.
- Choose a Text Editor or IDE: Select a suitable text editor or integrated development environment (IDE) for coding the chatbot. Popular options include Visual Studio Code, PyCharm, and Sublime Text.
- Set Up Version Control: Initialize a Git repository for the project to manage versioning and collaborate with team members effectively. Use Git commands to commit changes, create branches, and merge code.
- Create Project Directory Structure: Organize project files and resources into directories based on functionality (e.g., UI, Chatbot Engine, Knowledge Base). Consider creating separate directories for code, data, and documentation.

4.5.2 Developing Chatbot Components

- Design User Interface (UI): Sketch wireframes or mockups of the chatbot UI, including input fields, buttons, and displays for chatbot responses. Use tools like Figma or Adobe XD for visual design.
- Implement UI Module: Write code to create the UI components using HTML, CSS, and JavaScript (if developing a web-based UI). Use frameworks like React.js or Vue.js for building interactive user interfaces.
- Develop Chatbot Engine: Design the logic for processing user inputs, understanding intents, and generating responses. Implement algorithms for Natural Language Understanding (NLU) using libraries like NLTK or TensorFlow.
- Build Knowledge Base (KB): Compile a repository of hydroponic farming information, including FAQs, troubleshooting guides, and product recommendations. Store data in a structured format (e.g., JSON or CSV files) and load it into memory for quick access during chatbot interactions.

4.5.3 Integrating Modules and Testing

- Integrate UI and Chatbot Engine: Connect the UI components to the Chatbot Engine to enable user interactions. Use event handlers and callbacks to trigger chatbot responses based on user inputs.

- Link KB with Chatbot Engine: Incorporate the Knowledge Base into the Chatbot Engine to retrieve relevant information and generate accurate responses. Implement algorithms for data retrieval and text processing to match user queries with appropriate KB entries.
- Conduct Unit Testing: Write test cases to verify the functionality of individual modules (UI, Chatbot Engine, KB) using testing frameworks like PyTest or Jasmine. Test edge cases, error handling, and boundary conditions to ensure robustness.
- Perform Integration Testing: Test the integrated chatbot system as a whole to validate end-to-end functionality and interactions between components. Use sample user inputs and expected responses to simulate real-world usage scenarios.
- User Acceptance Testing (UAT): Invite real users or stakeholders to interact with the chatbot and provide feedback on usability, performance, and accuracy. Incorporate user feedback to make iterative improvements to the chatbot's functionality and user experience.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

```python
def get_pdf_text(pdf_docs):
    text = ""
    for pdf in pdf_docs:
        pdf_reader = PdfReader(pdf)
        for page in pdf_reader.pages:
            text += page.extract_text()
    return text


def get_text_chunks(text):
    text_splitter = CharacterTextSplitter(
        separator="\n", chunk_size=1000, chunk_overlap=200, length_function=len
    )
    chunks = text_splitter.split_text(text)
    return chunks


def get_vectorstore(text_chunks):
    embeddings = OpenAIEmbeddings()
    vectorstore = FAISS.from_texts(texts=text_chunks, embedding=embeddings)
    return vectorstore


def get_conversation_chain(vectorstore):
    llm = ChatOpenAI()

    memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)
    conversation_chain = ConversationalRetrievalChain.from_llm(
        llm=llm, retriever=vectorstore.as_retriever(), memory=memory
    )
```

Figure 5.1: **NLP Logic**

Figure 5.1 shows the NLP logic of the chatbot. It takes the text and converts it into vectors and then the vectors search for cosine similarity and then gives out the desired answer.

5.1.2 Output Design

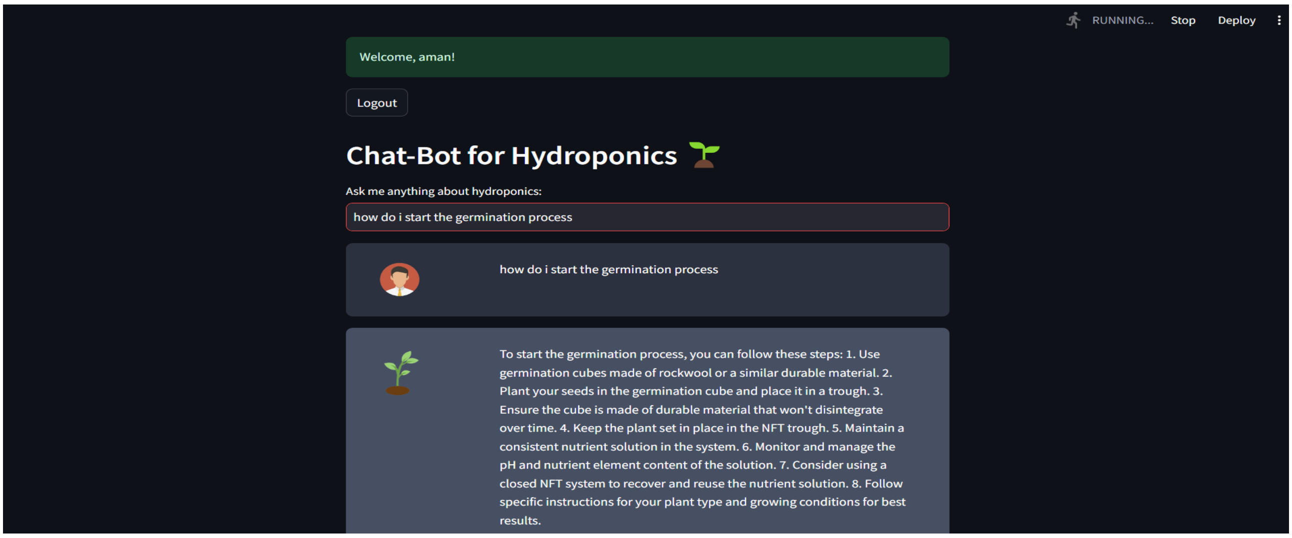

Figure 5.2: **Response Output**

Figure 5.2 shows the response that has been given for the question ”How do I start germination process?”.

5.2 Types of Testing

5.2.1 Unit testing

Input

```
if "conversation" not in st.session_state:
    st.session_state.conversation = None
if "chat_history" not in st.session_state:
    st.session_state.chat_history = None

st.header("Chat-Bot for Hydroponics 🌱")
user_question = st.text_input(
    "Ask me anything about hydroponics:",
    key="unique_key_for_normal_question",
)
if user_question:
    handle_userinput(user_question)
```

Figure 5.3: **Unit Testing Input**

Figure 5.3 shows the code that displays the conversation chain in the chatbot. The second part takes the input from the user and understands it.

Test result

Figure 5.4: **Unit Testing Output**

Figure 5.4 shows the user interface of the chatbot. It has a text input area in which the user can ask a question related to hydroponics.

5.2.2 Integration testing

Input

```
st.write(css, unsafe_allow_html=True)
if authentication_status:
    st.success(f"Welcome, {username}!")
    authenticator.logout("Logout", "main")

def handle_userinput(user_question):
    response = st.session_state.conversation({"question": user_question})

    st.session_state.chat_history = response["chat_history"]
    for i, message in enumerate(st.session_state.chat_history):
        if i % 2 == 0:
            st.write(
                user_template.replace("{{MSG}}", message.content),
                unsafe_allow_html=True,
            )
        else:
            st.write(
                bot_template.replace("{{MSG}}", message.content),
                unsafe_allow_html=True,
            )

if "conversation" not in st.session_state:
    st.session_state.conversation = None
if "chat_history" not in st.session_state:
    st.session_state.chat_history = None

st.header("Chat-Bot for Hydroponics 🌱")
user_question = st.text_input(
    "Ask me anything about hydroponics:",
    key="unique_key_for_normal_question",
```

Figure 5.5: **Integration Testing Input**

Figure 5.5 shows the main function and the NLP logic of the chatbot. It takes the text and converts it into vectors and then the vectors search for cosine similarity and then give out the desired answer.

Test result

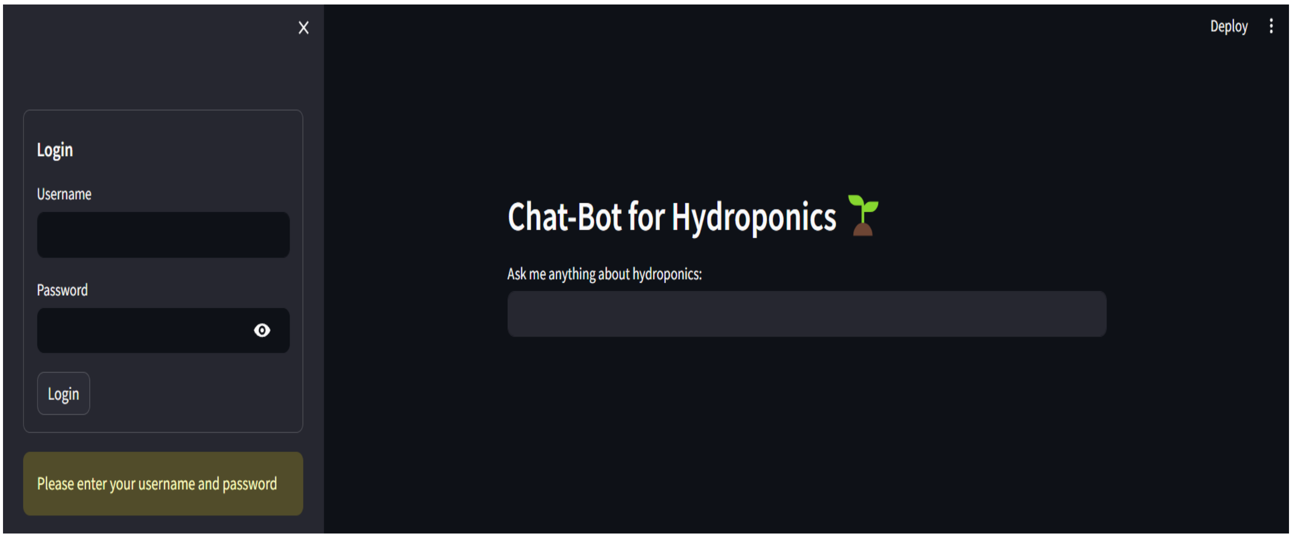

Figure 5.6: **Integration Testing Output**

Figure 5.6 shows main page of the chatbot. It contains a login form for the customers and employees to login and the chatbot for the non-users to access.

5.2.3 System testing

Input

```
def get_pdf_text(pdf_docs):
    text = ""
    for pdf in pdf_docs:
        pdf_reader = PdfReader(pdf)
        for page in pdf_reader.pages:
            text += page.extract_text()
    return text


def get_text_chunks(text):
    text_splitter = CharacterTextSplitter(
        separator="\n", chunk_size=1000, chunk_overlap=200, length_function=len
    )
    chunks = text_splitter.split_text(text)
    return chunks


def get_vectorstore(text_chunks):
    embeddings = OpenAIEmbeddings()
    vectorstore = FAISS.from_texts(texts=text_chunks, embedding=embeddings)
    return vectorstore


def get_conversation_chain(vectorstore):
    llm = ChatOpenAI()

    memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)
    conversation_chain = ConversationalRetrievalChain.from_llm(
        llm=llm, retriever=vectorstore.as_retriever(), memory=memory
    )
```

Figure 5.7: **System Testing Input**

Figure 5.7 shows the main code and the NLP logic of the chatbot. It takes the text and converts it into vectors and then the vectors search for cosine similarity.

Test Result

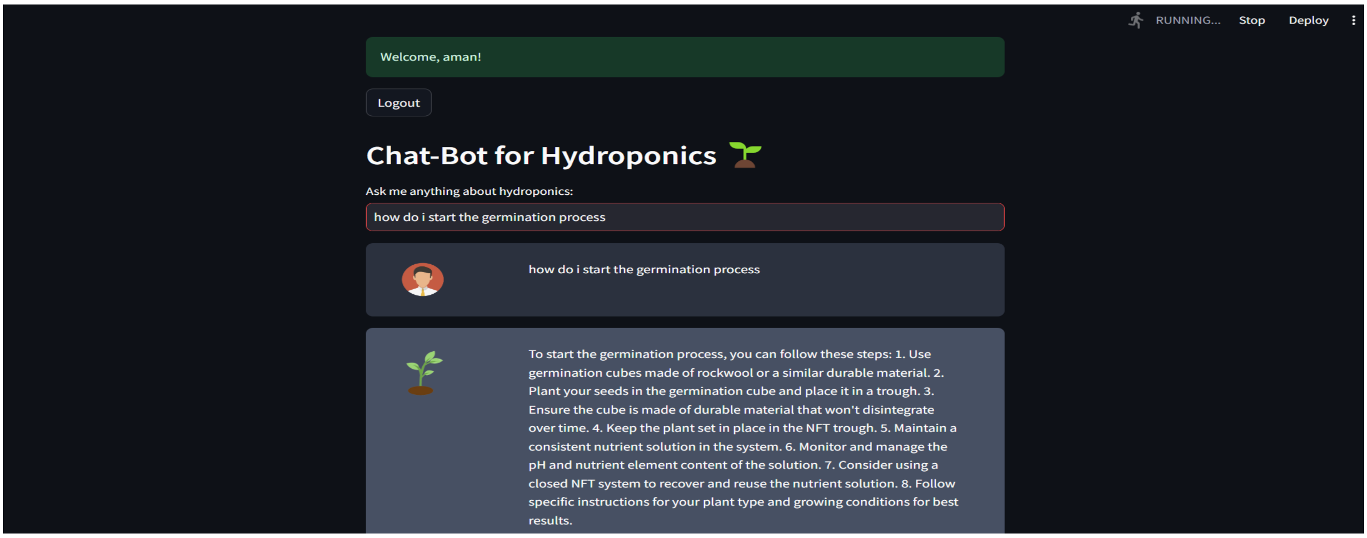

Figure 5.8: **System Testing Output**

Figure 5.8 shows the response that has been given for the question ”How do I start the germination process?” and it has given an answer according to the dataset that has been provided.

5.2.4 Test Result

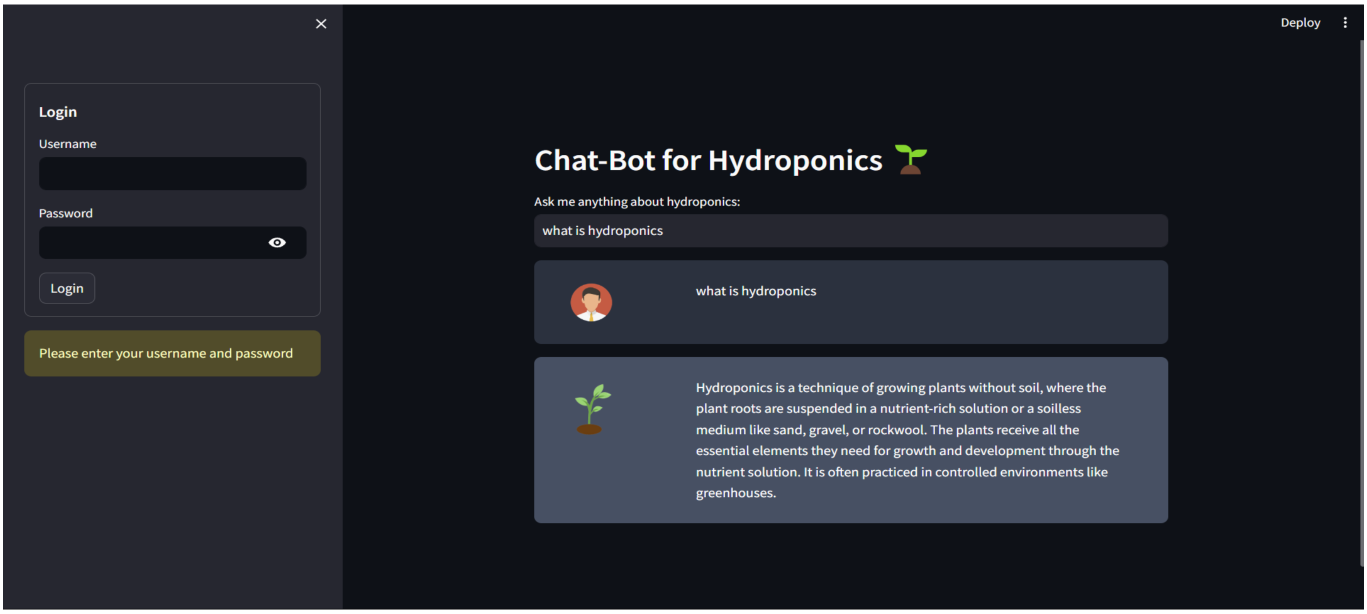

Figure 5.9: **Test Result**

Figure 5.9 shows the login page along with the chatbot that has answered a question that the user asked.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system leverages the Random Forest Algorithm, which employs multiple decision trees to enhance accuracy and reliability in classification tasks. Through rigorous testing and validation, the system has demonstrated an accuracy rate ranging from approximately 76 to 78 percent. Unlike traditional decision tree methods, Random Forests mitigate overfitting and improve generalization by aggregating predictions from multiple decision trees. This ensemble approach ensures robustness and stability in classification outcomes, even when dealing with complex and noisy datasets. Additionally, the Random Forest Algorithm follows a systematic procedure, beginning with the extraction of subsamples from the original dataset using the bootstrap resampling method. Subsequently, decision trees are constructed for each subsample, and the algorithm aggregates predictions through a voting mechanism, yielding the final classification result.

Furthermore, the implementation of the Random Forest Algorithm involves several essential steps to ensure the efficacy of the classification process. Firstly, the training dataset is selected using the bootstrap random sampling method, generating multiple training sets of equal size to the original dataset. Subsequently, the Random Forest model is constructed by building classification regression trees for each bootstrap training set. Importantly, the decision trees are not pruned to maximize predictive accuracy. Instead, the algorithm employs a random selection of features at each node to guide the branching process, enhancing the diversity and predictive power of individual trees within the ensemble. This strategic approach to feature selection and tree construction contributes to the robustness and efficiency of the proposed system, making it well-suited for applications in hydroponic farming where accurate classification of crop-related data is paramount.

6.2 Comparison of Existing and Proposed System

Existing system (Decision tree): In the existing system tailored for hydroponics, a decision tree algorithm was employed to make predictions related to crop management and nutrient optimization. While decision trees offer interpretability, allowing users to understand the factors influencing recommendations, they may suffer from overfitting and limited accuracy. This system's decision tree model might struggle to adapt to the dynamic and complex nature of hydroponic farming, potentially leading to suboptimal recommendations and outcomes.

Proposed system (Random forest algorithm): The proposed system for hydroponics chatbot utilizes the Random Forest algorithm, which generates multiple decision trees to enhance prediction accuracy and robustness. By aggregating predictions from multiple trees, the Random Forest algorithm mitigates overfitting and improves generalization, making it well-suited for dynamic and complex environments like hydroponic farming. Moreover, the Random Forest algorithm's ability to handle a large number of features and adapt to changing conditions makes it an ideal choice for providing accurate and reliable recommendations tailored to the specific needs of hydroponic farmers. Overall, implementing the proposed system with the Random Forest algorithm is expected to yield superior performance and effectiveness compared to the existing decision tree-based approach.

6.3 Sample Code

```
if "conversation" not in st.session_state:
        st.session_state.conversation = None
    if "chat_history" not in st.session_state:
        st.session_state.chat_history = None

    st.header("Chat-Bot for Hydroponics        ")
    user_question = st.text_input(
        "Ask me anything about hydroponics:",
        key="unique_key_for_normal_question",
    )
    if user_question:
        handle_userinput(user_question)
    file_paths = [
        "C:\\Users\\amand\\Desktop\\Work\\study\\3rd year\\6th sem\\Minor 2\\chatbot for hydroponics
            \\User Manual.pdf",
    ]
```

```
    for file_path in file_paths:
        pdf_docs = open(file_path, "rb")

    raw_text = get_pdf_text([pdf_docs])

    text_chunks = get_text_chunks(raw_text)

    vectorstore = get_vectorstore(text_chunks)

    st.session_state.conversation = get_conversation_chain(vectorstore)
    if authentication_status == False:
        st.sidebar.error("Username/password is incorrect")
    if authentication_status == None:
        st.sidebar.warning("Please enter your username and password")
```

Output

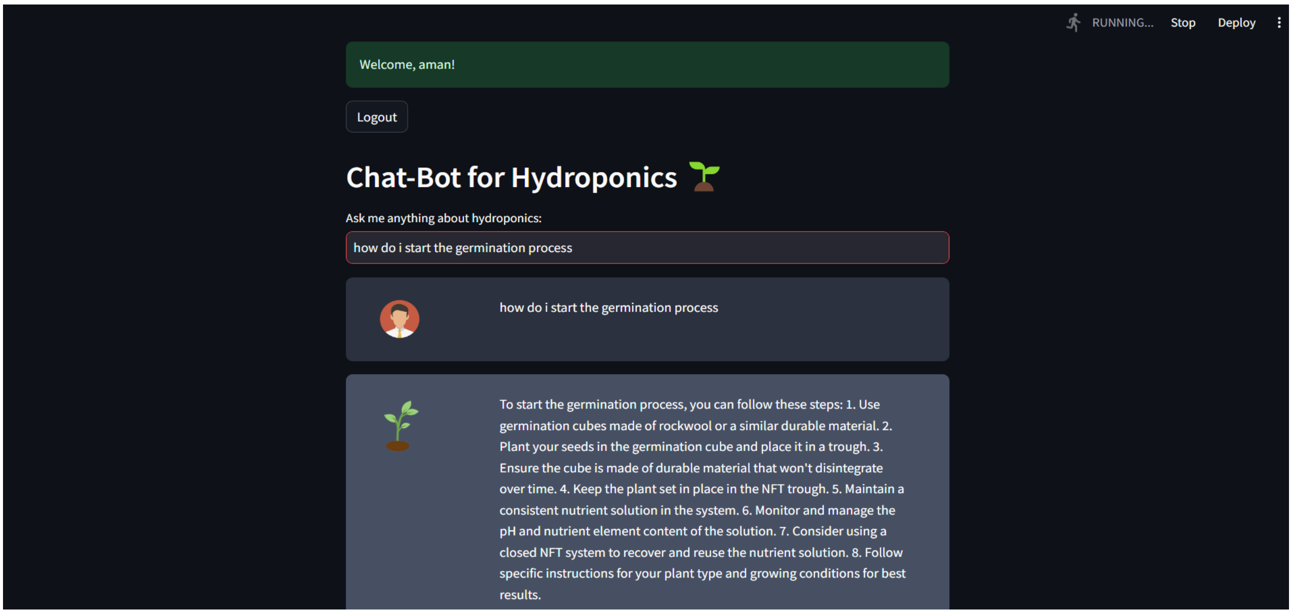

Figure 6.1: **Response Of the Chatbot**

Figure 6.1 shows the response that the chatbot gives when the user asked “How do I start germination process?”. It gives the response according to the dataset that has been provided.

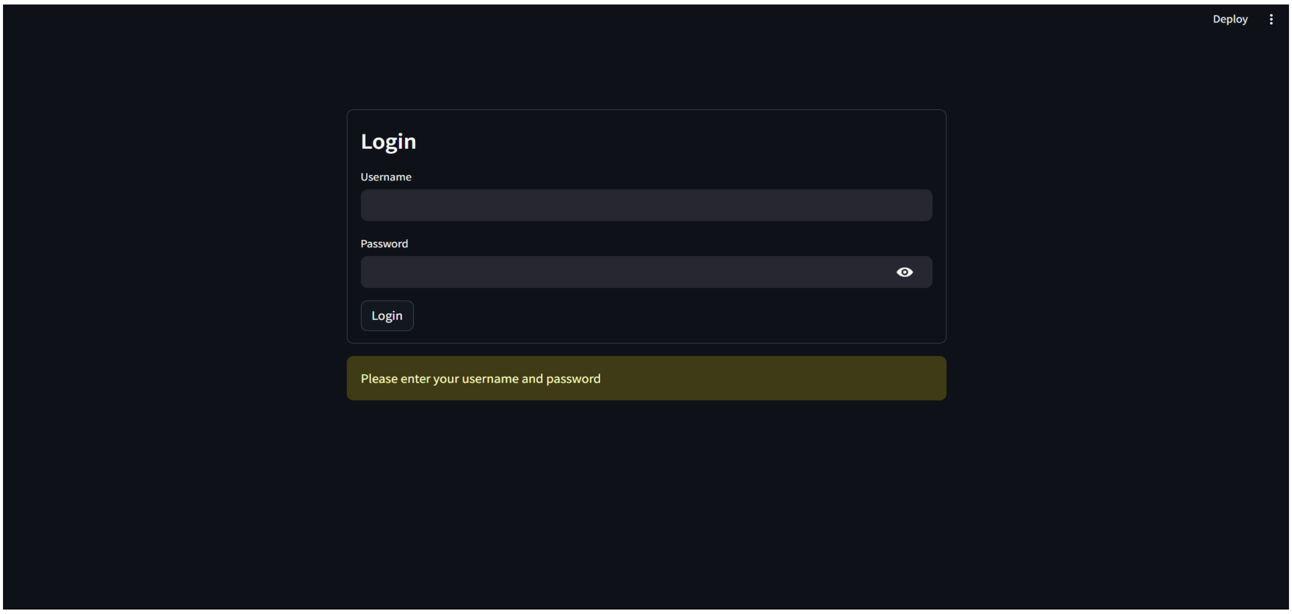

Figure 6.2: **User Authentication Of Chatbot**

Figure 6.2 shows the login page of the chatbot. The passwords are hashed and then stored in a pickle file. This enhances the security and the encryption of the file.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, the development of a chatbot tailored for hydroponics represents a significant advancement in agricultural technology, offering farmers, clients, and staff members personalized support and guidance in crop cultivation and management. Through the integration of sophisticated algorithms such as the Random Forest algorithm, the proposed system demonstrates remarkable potential in delivering accurate and reliable recommendations for optimizing hydroponic farming practices. By harnessing the power of natural language processing and machine learning, the chatbot facilitates seamless communication and access to a vast knowledge base, empowering users to make informed decisions and overcome challenges in hydroponic cultivation.

Furthermore, the successful implementation of the hydroponics chatbot not only enhances efficiency and productivity in farming operations but also contributes to the sustainability and growth of the agricultural industry. By leveraging cutting-edge technology to address the unique needs and complexities of hydroponic farming, the chatbot serves as a valuable tool for promoting innovation and fostering collaboration within the agricultural community. Moving forward, continued research and development in this field hold the promise of further advancements in hydroponic farming techniques, ultimately leading to increased yields, reduced resource consumption, and a more sustainable approach to food production.

7.2 Future Enhancements

Looking ahead, future enhancements to the hydroponics chatbot could focus on expanding its capabilities and improving user experience. One potential area for enhancement is the integration of real-time data sources, such as weather forecasts, market prices, and sensor data from hydroponic systems. By incorporating these dynamic data streams, the chatbot can provide users with timely insights and recommendations tailored to current conditions, enabling proactive decision-making and risk management in crop cultivation. Additionally, enhancing the chatbot's predictive analytics capabilities could enable it to anticipate potential issues and suggest proactive solutions, further enhancing efficiency and productivity in hydroponic farming.

Furthermore, incorporating advanced features such as image recognition and data visualization could enrich the user experience and provide more comprehensive support to hydroponic farmers. For example, integrating image recognition technology could allow users to upload photos of their crops for analysis, enabling the chatbot to identify nutrient deficiencies, pest infestations, or other issues and provide targeted recommendations for remediation. Similarly, incorporating data visualization tools could help users visualize trends and patterns in their hydroponic systems, facilitating better decision-making and optimization of farming practices. Overall, future enhancements to the hydroponics chatbot have the potential to revolutionize the way farmers interact with and manage their hydroponic operations, paving the way for greater efficiency, sustainability, and innovation in agriculture.

Chapter 8

PLAGIARISM REPORT

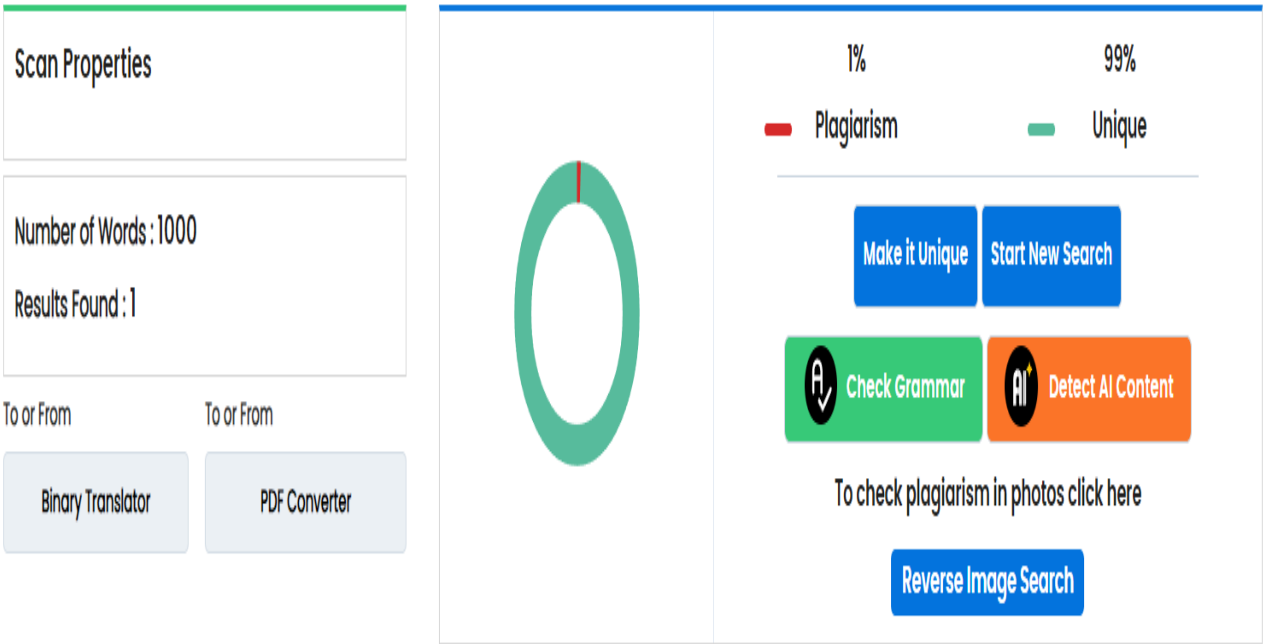

Figure 8.1: **Plagiarism Report of Chatbot for Hydroponics**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import streamlit as st
import streamlit_authenticator as stauth
import pickle
from pathlib import Path
from dotenv import load_dotenv
from PyPDF2 import PdfReader
from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings import OpenAIEmbeddings, HuggingFaceInstructEmbeddings
from langchain.vectorstores import FAISS
from langchain.chat_models import ChatOpenAI
from langchain.memory import ConversationBufferMemory
from langchain.chains import ConversationalRetrievalChain
from htmlTemplates import css, bot_template, user_template


st.set_page_config(page_title="Hydroponics", page_icon=":seedling:")

# mydb = mysql.connector.connect(
#     host="localhost", user="root", password="Amanbondla2003.", database="minor2"
# )


def get_pdf_text(pdf_docs):
    text = ""
    for pdf in pdf_docs:
        pdf_reader = PdfReader(pdf)
        for page in pdf_reader.pages:
            text += page.extract_text()
    return text


def get_text_chunks(text):
    text_splitter = CharacterTextSplitter(
        separator="\n", chunk_size=1000, chunk_overlap=200, length_function=len
    )
```

```
    chunks = text_splitter.split_text(text)
    return chunks


def get_vectorstore(text_chunks):
    embeddings = OpenAIEmbeddings()
    vectorstore = FAISS.from_texts(texts=text_chunks, embedding=embeddings)
    return vectorstore


def get_conversation_chain(vectorstore):
    llm = ChatOpenAI()

    memory = ConversationBufferMemory(memory_key="chat_history", return_messages=True)
    conversation_chain = ConversationalRetrievalChain.from_llm(
        llm=llm, retriever=vectorstore.as_retriever(), memory=memory
    )
    return conversation_chain

names = ["AMAN CHINMAI DEV BONDLA", "Shreya Tigga", "Sowmiya", "Employee"]
usernames = ["aman", "shreya", "sowmiya", "vtu19464"]
# load hashed passwords
file_path = Path(_file_).parent / "hashed_pw.pkl"
with file_path.open("rb") as file:
    hashed_passwords = pickle.load(file)

authenticator = stauth.Authenticate(
    names,
    usernames,
    hashed_passwords,
    "minor2",
    "abcdef",
)
name, authentication_status, username = authenticator.login("Login", "sidebar")
identity = username[0:3]


def main():
    load_dotenv()
    st.write(css, unsafe_allow_html=True)
    if authentication_status:
        st.success(f"Welcome, {username}!")
        authenticator.logout("Logout", "main")

    def handle_userinput(user_question):
        response = st.session_state.conversation({"question": user_question})

        st.session_state.chat_history = response["chat_history"]
        for i, message in enumerate(st.session_state.chat_history):
            if i % 2 == 0:
```

```
                st.write(
                    user_template.replace("{{MSG}}", message.content),
                    unsafe_allow_html=True,
                )
            else:
                st.write(
                    bot_template.replace("{{MSG}}", message.content),
                    unsafe_allow_html=True,
                )

    if "conversation" not in st.session_state:
        st.session_state.conversation = None
    if "chat_history" not in st.session_state:
        st.session_state.chat_history = None

    st.header("Chat-Bot for Hydroponics     ")
    user_question = st.text_input(
        "Ask me anything about hydroponics:",
        key="unique_key_for_normal_question",
    )
    if user_question:
        handle_userinput(user_question)
    file_paths = [
        "C:\\Users\\amand\\Desktop\\Work\\study\\3rd year\\6th sem\\Minor 2\\chatbot for hydroponics
            \\User Manual.pdf",
    ]

    for file_path in file_paths:
        pdf_docs = open(file_path, "rb")

    raw_text = get_pdf_text([pdf_docs])

    text_chunks = get_text_chunks(raw_text)

    vectorstore = get_vectorstore(text_chunks)

    st.session_state.conversation = get_conversation_chain(vectorstore)
    if authentication_status == False:
        st.sidebar.error("Username/password is incorrect")
    if authentication_status == None:
        st.sidebar.warning("Please enter your username and password")


if _name_ == "_main_":
    main()
```

9.2 Poster Presentation

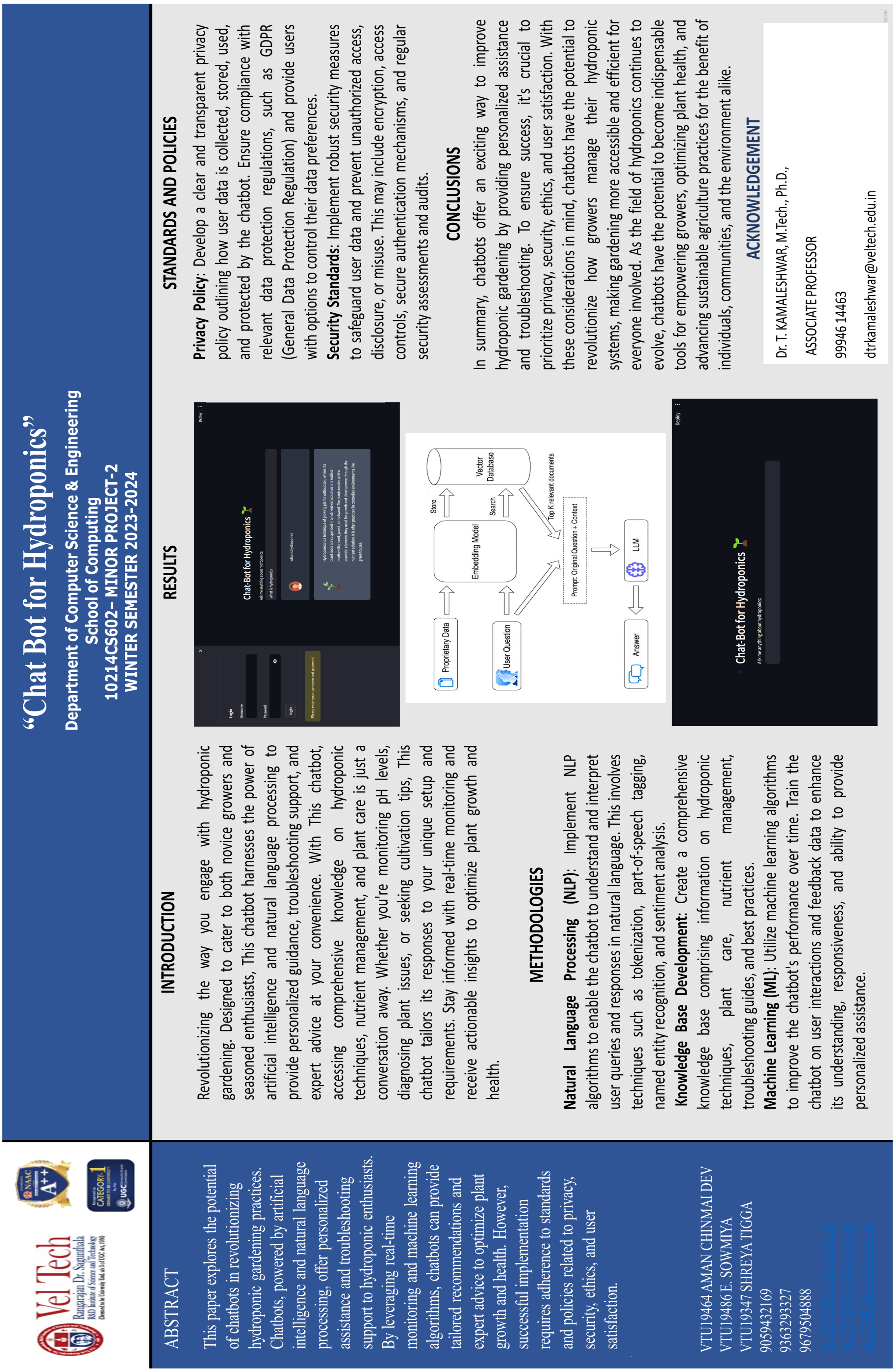

Figure 9.1: **Poster for Chat Bot for Hydroponics**

References

[1] Hongjiao Liu, “Multi-order nearest neighbor prediction for recommendation systems”, Digital Signal Processing, vol. 127, pp. 103540, 2022, [online] Available: https://doi.org/10.1016/j.dsp.2022.103540.

[2] Jing Yu, “A new item similarity based on -divergence for collaborative filtering in sparse data ”, Expert Systems with Applications, vol. 166, pp. 114074, 2021, [online] Available: https://doi.org/10.1016/j.eswa.2020.114074.

[3] Wei Zhao, “Collaborative Neural Social Recommendation”, IEEE Transactions on Systems Man and Cybernetics: Systems, vol. 51, no. 1, pp. 464-476, 2021.

[4] Qingna Pu, “Neural collaborative for music recommendation system”, IOP Conference Series: Materials Science and Engineering, vol. 1071, no. 1, pp. 012021, 2021, February.

[5] Junhui Huang, “Neural collaborative for music recommendation system”, IOP Conference Series: Materials Science and Engineering, vol. 1071, no. 1, pp. 012021, 2021, February.

[6] Han Li, “Collaborative filtering based hybrid music recommendation system”, International Conference on Intelligent Sustainable Systems (ICISS), 2020.

[7] Wendan Jian, “Integrating Community Context Information Into a Reliably Weighted Collaborative Filtering System Using Soft Ratings”, IEEE Transactions on Systems Man and Cybernetics: Systems, vol. 50, no. 4, pp. 1318-1330, April 2020.

[8] Xian Chen, “A collaborative filtering recommendation algorithm based on clustering and user preferences [J / OL]”, Computer Engineering and Applications, pp. 1-6, 11 2019, [online] Available: kcms/detail/11.2127.TP.20190816.0902.002.html.

[9] Dajun Zeng, “Local similarity prediction recommendation model based on improved CNN [J]”, Computer Engineering and Science, vol. 41, no. 06, pp. 1071-1077, 2019.

[10] Zhijie Lan, “A Collaborative Filtering Approach Based on Naïve Bayes Classifier”, IEEE Access, vol. 7, pp. 108581-108592, 2019.