

# **GuidoBot : A Novel Approach for Assisting Faculty in Tracking Goals and Finding Tasks**

*A Project Report submitted in partial fulfilment*

*of the requirements for the award of the degree of*

**Bachelor of Technology**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**170030875**

**NAGOTHU SRUJAN**

**170031326**

**UTLAPALLI MAHESH**

**170031391**

**VEMURI POOJITHA**

**170030161**

**BONELA SYAM JASON**

*Under the esteemed guidance of*

**J. SURYA KIRAN** *M.Tech. (Ph. D.)*

*Assistant Professor*



**Koneru Lakshmaiah Education Foundation**

(Deemed to be University estd., u/s 3 of UGC Act 1956)

Greenfields, Vaddeswaram, Guntur (Dist.), Andhra Pradesh - 522502

November, 2019

**K L (Deemed to be) University**  
**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



**DECLARATION**

The Major Project Report entitled “**GuidoBot : A Novel Approach for Assisting Faculty in Tracking Goals and Finding Tasks**” is a record of bonafide work of Nagothu Srujan-170030875, Utlapalli Mahesh – 170031326, Vemuri Poojitha - 170031361 and Bonela Syam Jason -17030161 submitted in partial fulfilment for the award of **B. Tech.** in “Computer Science and Engineering” year 2020-2021.

<b>170030875</b>	<b>Nagothu Srujan</b>
<b>170031326</b>	<b>Utlapalli Mahesh</b>
<b>170031391</b>	<b>Vemuri Poojitha</b>
<b>170030161</b>	<b>Bonela Syam Jason</b>

**K L (Deemed to be) University**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



## **CERTIFICATE**

This is to certify that the project report entitled “GuidoBot : A Novel Approach for Assisting Faculty in Tracking Goals and Finding Tasks” is a record of bonafide work of Nagothu Srujan-170030875, Utlapalli Mahesh – 170031326, Vemuri Poojitha - 170031361 and Bonela Syam Jason -17030161 submitted in partial fulfilment for the award of B. Tech. in “Computer Science and Engineering” year 2020-2021.

**Mr. J. SURYA KIRAN**  
**PROJECT GUIDE**  
DEPARTMENT OF CSE  
KLEF

**Mr. HARI KIRAN VEGE**  
**HEAD OF THE DEPARTMENT**  
DEPARTMENT OF CSE  
KLEF

**EXTERNAL EXAMINER**

### **ACKNOWLEDGEMENT**

Our sincere thanks to the project guide, **J. Surya Kiran** for his outstanding support throughout the project for the successful completion of the work.

We express our gratitude to **Hari Kiran Vege**, Head of the Department for Computer Science and Engineering, CSE for providing us with adequate facilities, ways, and means by which we can complete this term paperwork.

We would like to place on record the deep sense of gratitude to the honorable Vice-Chancellor, K L University for providing the necessary facilities to carry the concluded project work. Last but not the least we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our work.

<b>170030875</b>	<b>Nagothu Srujan</b>
<b>170031326</b>	<b>Utlapalli Mahesh</b>
<b>170031391</b>	<b>Vemuri Poojitha</b>
<b>170030161</b>	<b>Bonela Syam Jason</b>

## **ABSTRACT**

In the world of Artificial intelligence, we have many inventions that facilitate human works, one of such innovation is a Chabot. We are developing a Window based Guido Bot as a guide which helps faculty for Goal tracking and in managing daily tasks effectively. We are designing this Chabot with three different components like Profile Maintenance module which deals with the academic profile, status of research publications of faculty. And an important module is a Goal Tracking module that helps faculty to achieve their academic goals, research goals with timely updates and notifications. And the Task Finding module helps faculty in assigning and tracking the tasks status with daily updates of pending tasks along with their deadlines. And we are developing a web application as an interface for admins (Management) to monitor and issue goals and administrative tasks for faculty. And the Guido Bot uses this interface as a data source for efficient performance

## **CONTENTS**

Certificate

Declaration

Acknowledgements

Abstract

<b>S.No</b>	<b>Chapters</b>	<b>Page Numbers</b>
1	Introduction	1
2	Problem Statement	3
3	Literature Survey	4
4	Existing System	9
5	Proposed System	10
6	Methodology	11
7	Block Diagram	14
8	Software Requirements	17
9	Experimental set up and results	20
10	Implementation	21
11	Output screens	47
12	Conclusion	53
13	References	54
14	Plagiarism Report	---
15	Paper Acceptance	---

## **1. INTRODUCTION**

A chatbot is one of the most important game-changing innovations in the field of AI and ML during the last decade. A Chatbot is a Virtual Enterprise software application with a Conversational User Interface (CUI) that provides diverse services in the fields of health, education, management, and Government services. It permits various types of interchanges between a human and a machine, by means of messages or voice order. A chatbot is modified to work freely from a human administrator. It can answer questions planned in regular language and react like a genuine individual. Chatbot revolution, hence brought communication, to an entirely new customized level. Today's chatbots, further developed since then, are ready to answer considerably complex questions [13] in an effective timely manner and able to automate human tasks efficiently. There are two types of Chatbots: i) Simple Chatbots – Chatbots that function on predefined rules and can answer to only limited queries. To find an answering user needs to make several selections [14]. If a user asks a question that doesn't match with the database present, the chatbot would be unable to answer the query and print a message "sorry I don't know" as these chatbot's inbuilt code could solve only simple use cases. ii) Smart Chatbots – Chatbots that rely on Machine Learning and Normal Language to know the user's language which is intelligent enough to remember conversations with specific users to learn and grow over time in a dynamic environment. Instead of static answers, these chatbots respond with several other recommendations based on the previous chat data and data from similar user personalities [15]. These chatbots are built on code that is complex in nature. Smart Chatbots have endless applications. With the help of technology, we had seen considerable innovation to track and manage individual and group tasks, i.e. through reminder apps, email, messenger, etc. In this current AI world, we have a wide variety of chatbot applications that infiltrate every corner in educational institutions that assisting Management and students for effective interactions and querying well about the institution in real-time. There is no existing system that provides assistance for faculty in order to achieve their goals in a timely manner. In a few Applications User's need to write complex queries in order to get the desired result or users need to refer to the web portal (ERP) every time which is time-consuming. Also, there is an ambiguity regarding timings scheduled for classes and events. So, to overcome this problem there is a need for a system that guides the faculty in order to create, assign, manage, track, and accomplish their goals and objectives in the desired time frame. chatbots are profoundly

accessible, incredibly refine, and can be a partner for faculty. Groups that take on chatbots will end up approaching up to effectively achieving the numerous impossible assignments and discrepant objectives that their work requests.



## **2. PROBLEM STATEMENT**

In the current AI world, we have a wide variety of chatbot applications that infiltrate every corner in educational institutions that assisting Management and students for effective interactions and querying well about institution in a real time. But there is a need of a system that guides the faculty and students in order to accomplish their goals and objectives in desired time frame. The proposed idea Guido Bot mainly focusing on faculty which will provide a chatbot assistance that will help faculty in effective tracking and accomplishing their goals with proper reminders and updates in timely manner.

### **3. LITERATURE SURVEY**

**i) Title: A Question Answering and Quiz Generation Chatbot for Education**

**Authors: A.S. Sreelakshmi, S.B. Abhinaya, Aishwarya Nair, S. Jaya Nirmala**

In this paper, Chatbot performs three tasks i.e. answering user questions, generating a quiz for the user, evaluating the performance of the user. The user provides documents as input to the Chatbot. This input undergoes several pre-processing [16] steps and acts as an expert system and this expert system is taken as input for the first module as well as the second module. The first module consists of ranking functions to take out the top 10 queries from the expert system by using the BM25 score. These queries are given as input to the neural network. This output of the neural network will be the answer given by the chatbot to the user questions. The second module ranks sentences based on their significance to the content in the expert system. Relevant sentences from the expert system are modified using NLP and are framed as questions to the user quiz. It is assessed based on the Answerability of a question and the Suitability of a sentence. Quiz answers given by the user are given as input to the third module i.e. Answer Comparison module. Its outputs score indicating the performance of the user Main drawbacks are feedback-based improvement is not present in order to generate the user level of understanding of the question. Every time user needs to upload the required documents/pdf's/pics to the chatbot. Complex Questions can be answered by consolidating the required data from all the uploaded pdfs.[1]

**ii) Title: Intelligent Chatbot for Guided Navigation of Repository Content**

**Authors: Anjali Mishra, Shruti Sapre, Shruti Shinde, S.N Shelke**

This paper presents an approach for Answering user questions for particular content. In the beginning, the user gets registered, and then after the successful registration process, the user uploads the file in NLP format. The uploaded file gets trained by the system and get saved in the database used in this application. Then the user can ask a question to the chatbot system that he/she want the answer and the user question is considered as an input. All the input related questions within the document are considered as streaming questions [17]. User input and streaming questions are taken into a queue. The system then selects the object from the user question and matches it with the streaming question by reading the file content and splitting the user input question line by line to match with the streaming questions and generate an

answer. The generated answer gets appended with the question and gets displayed to the user. The proposed system can be used by farmers and also at soil investigation centers. The main drawback is the document type is only one specific type and this application can be used by only one user at a time. [2]

**iii) Title: Adoption of AI-Chatbots to Enhance Student Learning Experience in Higher Education in India**

**Authors: Nitirajsingh Sandu, Ergun Gide**

This paper discusses how Chatbot technology can improvise student learning as in the same classroom, each student has different learning needs and interests. Therefore, each one can use the help of a specialized guide. It is common for students at all levels to use some kind of messaging service to communicate with each other and, occasionally, with their teachers but they are not available all the time. Students recognized that Chatbots respond in a faster mode as they are available 24/7. All the educational issues faced by the students would get clarified easily in a short span of time by using chatbots. They gave the highest response time when compared [18]. It also helped students in paying their fees. Immense benefits can be achieved by the students with the usage of a chatbot for higher education i.e., Student-centered learning. On the other hand, users also had a doubt of losing personal information when submitted to the chatbot when required. Users received inappropriate advice from the chatbot as sometimes the chatbot could not process the user input properly. User addiction to the chatbot might create a negative effect. [3]

**iv) Title: Chatbot and bullyfree chat**

**Authors: V. Selvi, S. Saranya, K. Chidida, R. Abarna**

It is impossible for humans to go through all the texts from conversations available online manually to determine if there is a problem. So, this paper discusses how NLP can be used in an efficient way to understand the user text and perform sentiment analysis to detect bully words. Sentiment analysis is generally denoted as techniques used to determine the predisposition of text, usually expressed in free text form. A sentence is parsed using NLP. Bully words are present in three forms i.e. in texts, user demography, and social networks features. Bully space is constructed. SDA concept is used to detect cyberbullying. A

supervised machine learning algorithm is used to check bot performance and continuously trained to enhance its performance. The advantage is that chatbot can get required information according to the user queries by automatic removal of bully words. The main drawback is the usage of the AdaBoost M1 algorithm as its performance is low when the noise in the dataset is high. [4]

**v) Title: Possibilities of used intelligence-based agents in instant messaging on e-government services**

**Authors: Dusko Sivcevic, Ivan Kosanin, Slobodan Nedeljkovic, Vojkan Nikolic**

This paper discusses how web agents can be used in both the individual, business, and all government sectors to offer better services. In this framework the deployment of artificial intelligence systems can provide transparency in communication to optimize government functionalities like the execution of ongoing policies, passing files to various sectors, delivering services, maintaining law & order. Automated Systems gives the highest response time to all the users. Applications are linked to instant messaging services like Viber allowing citizens to get address their issues. The benefits of web agents in governance are It simplifies the process of assembling and collecting government information regarding any department to the citizens and business. Ease of availability of police services 24\*7 for every citizen through online Agents. [5]

**vi) Title: Artificial intelligence in warehouse automation for flexible material handling**

**Authors: Hendrik Thamer, Axel Börold, Ariandy Yoga Benggolo, Michael Freitag**

This paper discusses how to improve transferability among the objects. Various sets of image samples of a particular class are considered, and these samples are trained to automatically detect how object class can contain its feature description. Based on this system can classify past unrecognized images and present familiar objects. With the help of deep learning [10] the system will be able to function in the right manner even though the user gives invalid inputs or in a dynamic environment. We can also overcome the problems regarding training data as the system now can identify all the parts of the image and multiple movements and positions of user images are considered. This approach can also be used for the objects presented in multiple platforms. Noise detection in the data can be improvised. [6]

**vii) Title: A Survey on Chat-Bot system for Agriculture Domain**

**Authors: Prashant Y. Niranjan, Vijay S. Rajpurohit, Rasika Malgi**

This paper discusses how chatbots can be implemented in the agriculture domain. Farmers can ask their questions related to weather changes, raw materials, plant protection methods [12], best practices adopted, etc. Farmers can utilize chatbots through mobile as it acts as an effective tool to provide efficient information access in a rural environment for people with less literacy rate and technology knowledge. Chatbot consists of 3 Modules. The first module is Question analysis second module is document processing [20] and answer extraction being the third module. Initially, the user query is processed using nlp techniques. In the second step i.e., document processing, all the query-related documents in the web are searched by using RNN (Recurrent neural network) as it consumes less time. The data set is in XML format. It consists of all the previous questions asked by the farmer indicating continuous learning by the chatbot.[7]

**viii) Title: A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation**

**Authors: Kyo-Joong Oh, Dongkun Lee, Byungsoo Ko, Ho-Jin Choi**

Today, the need for mental health [11] services is increased. This paper introduces a chatbot that can be used for mental healthcare. It recommends that continuous emotional recognition and the ethical response of the user need to be observed. It helps in notifying the user status when emergency situations occur. Chatbot gives the quickest response to the user's family members when required and can help in saving a life. Continuous Observations for a critical disease are not noted by the chatbot but helps the users in reducing their mood swings in the form of dialogues. people in distress can reach out. The need for one-on-one, individualized services have replaced other platforms. The user who needs psychiatric counseling can input contents to the chatbot in the form of text or the user can send images, videos, voice messages, etc. It checks the emotional imbalance as well as user problems and stores it in its log records of the user and gets updated continuously.[8]

**ix) Title: Chat bot Using API: Human To Machine Conversation**

**Authors: Sunil Punjabi, Vighnesh Sethuram, Vignesh Ramachandran, Ronit boddu, Shivshankar Ravi**

The main motive of this paper is to resolve the issues faced by the people using the IRCTC website. Input by the user is taken in the form of text or voice and is processed using NLP and accurate information is provided to the user regarding IRCTC. All the people using the site for ticket booking, checking seat availability, PNR status can get their queries solved quickly. All the services of the IRCTC website are provided under one roof. The chatbot is efficiently worked by combining with AI lead platforms like Google assistant and social media platforms like Facebook, etc. It is user-friendly as it can be used through any social networking site without any external installations. Response time of chatbot is high compared to MakeMyTrip bot. There are 4 phases in this Chatbot. Initially, dialog flow is used for creating entities and actions for the system. In the next phase with the help of java server-side coding is implemented. Further postman is used for testing and making a post request. Lastly, railway site API is used to provide accurate output.[9]

## **4. EXISTING SYSTEM**

The need of chatbot systems has become important with the ubiquitous use of personal systems which wish to communicate and the desire of their makers to provide natural language for them to communicate with humans. Nowadays, we see the chat bots everywhere Chat bots are the source of answers to the user's questions in any particular domain where it is operating. There are many applications that are incorporating a human appearance and intending to simulate human dialog, but in most of the cases the knowledge of the conversational bot is stored in a database created by a human experts.

### **4.1 Disadvantages of Existing System**

- **Lack of assistance:** There is no existing system as assistance for faculty.
- **Complex Querying:** User's need to write complex queries in order to get desired result.
- **Time Consuming:** In order to check the deadlines of objectives need to be done, User's need to refer the web portal (ERP) every time.
- Ambiguity regarding timings scheduled for classes and events

## **5. PROPOSED SYSTEM**

In the Proposed System we are developing a window-based Guido Bot as an assist that helps faculty in Goal Tracking and in managing daily tasks effectively. We are designing this Chatbot with three different modules like Profile Management which deals with academic and research Information of faculty. Goal tracking module helps to accomplish research, and Academic goals of faculty. Based upon role, goal tracking module has different functionalities i.e., based upon role of Assigner and Assignee. Task Finding module helps to accomplish administrative tasks, academic and research tasks of faculty.

### **5.1 Advantages of Proposed System**

- **Tracking Assistance:** we provide a chatbot assistance for faculty in tracking their goals.
- **Notifications:** Faculty will receive timely updates and reminders as notifications.
- **Time Efficient:** User can easily get his information about goals and tasks in no time.
- **No Ambiguity:** Faculty can easily manage their tasks and goals without any ambiguity with deadlines.
- Easy Interaction with human level Language.



## **6. METHODOLOGY**

Guido Bot framework consists of 3 modules

- i. Profile Management
- ii. Goal Tracking
- iii. Task Finding

Module 1: In Profile Management faculty can view their academic and research Information along with their awards and recognitions, and there is also a provision for them to update their profile using this Guido Bot.

Module 2: Goal tracking module helps to accomplish research, and Academic goals of faculty. Based upon role, goal tracking module has different functionalities i.e., based upon role of Assigner and Assignee.

For Assignee, he/she can view his/her pending goals and their deadlines and can upload the work that he/she completed as a progress.

For Assigner, he/she can verify the work of assignee and can update their progress of the goals based on his/her completion of goal.

Module 3: Task Finding module helps to accomplish administrative tasks, academic and research tasks of faculty.

For Assignee, he/she can view his/her pending tasks and their deadlines and can update their task status (InProgress / Completed) and can view the acceptance of their task from the assigner.

For Assigner, he/she can assign the tasks and able to track their status also have a privilege to accept or reject the task done by assignee.

---

**Algorithm 1** GoalTracking

---

**INPUT:**Set of Goals Assigned (G),Id of Goal (ID ) to be updated  
**OUTPUT:**Goal status (S)

Initialize: G = Goals Assigned  
P = Progress of Goal ( $0 \leq P \leq 100$ )  
S = Status of goal  
A = Acceptance of Assignee  
ID = Id of goal to be Update

```
1: procedure GOAL(G,ID):
2:   for each G do
3:     if (Id of G  $\equiv$  ID) then
4:       A  $\leftarrow$  Acceptence
5:       if (A  $\equiv$  "Accepted") then
6:         Uploading files of completed Work
7:         P  $\leftarrow$  Progress by Assigner
8:         if (P  $\equiv$  100) then
9:           S  $\leftarrow$  "Completed Goal"
10:        else if (P  $\leq$  99) then
11:          S  $\leftarrow$  "Under Progress"
12:        end if
13:      else if (A  $\equiv$  "Rejected") then
14:        S  $\leftarrow$  "Rejected Goal"
15:        break;
16:      end if
17:    else
18:      Continue;
19:    end if
20:  end for
21:  return S
```

---

Fig 1: Algorithm for Goal Tracking.

### **Algorithm 1: Goal Tracking**

Step 1: Among all the goals (G) assigned to him, assignee needs to select a particular goal using its goal id (Gid).

Step 2: If he accepts to work on that goal then he needs to change its status to accepted.

Step 3: He need to upload all the work that he progressed as a file. Assigner will verify it and then update the progress.

Step 4: If the progress reaches to 100% then the goal will be considered as completed else it will be considered as under progress.

Step 5: If he rejects to work on that goal then he needs to change the status to rejected.

---

**Algorithm 2** TaskFinding

---

**INPUT:**Set of Tasks Assigned (T),Id of Task (ID ) to be updated

**OUTPUT:**Task status (S)

Initialize: T = Assigned Tasks

S = Status of Task

A =Approval by Assigner

U= Updated task state by Assignee

ID = Id of Task to be updated

```
1: procedure TASK(T,ID):
2:   for each T do
3:     if (Id of T ==ID) then
4:       U ← Update Task state
5:       if (U ≡ "In Progress") then
6:         S ← "InProgress"
7:       else if (U ≡ "Completed") then
8:         S ← "Completed"
9:       if (S ≡ Completed) then
10:        A ← Assigner Approval
11:        if (A ≡ "Approved") then
12:          S ← "Task Approved"
13:        else if A ≡ "Rejected" then
14:          S ← "Task Rejected"
15:        end if
16:      end if
17:    end if
18:  else
19:    continue;
20:  end if
21: end for
22: return S
```

---

Fig 2: Algorithm for Task Finding.

### **Algorithm 2: Task Finding**

Step 1: Among all the tasks (T) assigned to him, assignee needs to select a particular task using its task id (Tid).

Step 2: Initially the status of the task is in assigned state and when assignee started working, he needs to change the status as InProgress.

Step 3: If he completes the task, he needs to update the status as completed.

Step 4: Once the status is in completed state the assigner will verify the task done by the assignee and either he can approve or reject the task submitted.

## 7. BLOCK DIAGRAM

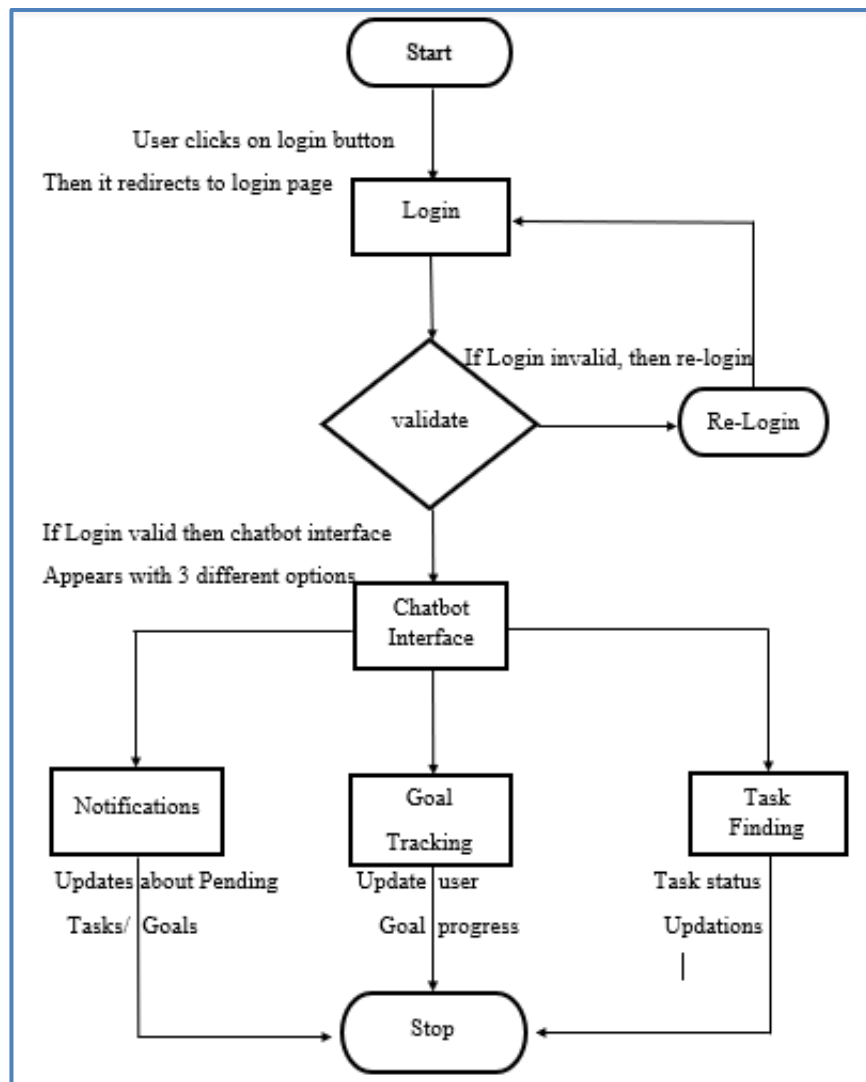


Fig 3: Block Diagram of Chatbot

### Workflow of the Guido Chatbot:

Step 1: If a Faculty is already a registered user, then he/she can sign in to the chatbot, or else the faculty need to get registered with the help of admin.

Step 2: When the Faculty sign in with the correct login credentials then they will be redirected to the Interface of the chatbot or else they will be redirected to the login page again.

Step 3: In the interface users have an option to open the Notifications, Chat, or Task Finding tab based on his/her interest.

Step 4: If the user opens Notifications tab then he/she is having different types of notifications that is pending goals and assigned tasks. Notifications can help user from not missing any of his tasks in his hectic schedule.

Step 5: If the user opens Chat tab then he/she can perform task updates, can change his/her goal progress that will be reflected with the admin, and manage their profile updates.

Step 6: If the user opens Task Finding tab then he/she can view his/her pending tasks and their deadlines and even he can update his task status using GuidoBot.

Step 7: Based on the user interest he/she can interact with the chatbot and can resolve all his queries instantly.

## 7.1 WEBSITE BLOCK DIAGRAM

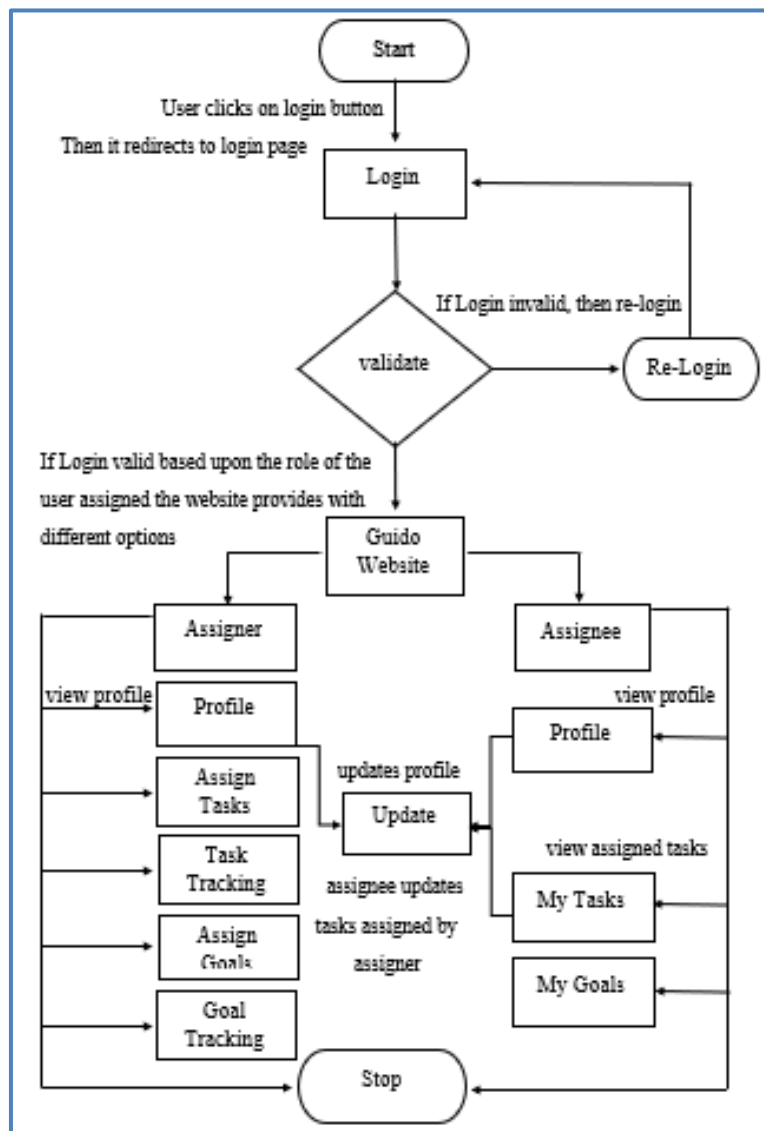


Fig 4: Block Diagram of Guido Website

## **Workflow of the Guido Website:**

Step 1: If a Faculty is already a registered user, then he/she can sign in to the chatbot, or else the faculty need to get registered with the help of admin.

Step 2: When the Faculty sign in with the correct login credentials then they will be redirected to the Home page (Landing Page) of the website or else they will be redirected to the login page again.

Step 3: Now based upon his/her role of the user assigned the Guido website provides different views, which is the Assignee view or Assigner view.

Step 4: If the Assignee is the user then the assignee is having different options in his/her dashboard.

- 1) Profile
- 2) Assign Tasks
- 3) Task Tracking
- 4) Assign Goals
- 5) Goal Tracking

a) In Profile he/she can view the profile and manage their profile updates.

b) In Assign Tasks he/she can assign tasks to the assigner.

c) In Task Tracking he/she can track the tasks that are already assigned by the assignee, and even change his/her goal progress.

d) In Assign Goals he/she can assign goals to the assigner.

e) In Goal Tracking he/she can track the goals that are already assigned by the assignee, and even change his/her goal progress.

Step 5: If the Assigner is the user then the assigner is having different options in his/her dashboard.

1. Profile
2. My Tasks
3. My Goals

f) In Profile he/she can view the profile and manage their profile updates.

g) In My Tasks he/she can check the tasks that are assigned by the assignee, and even change his/her goal progress.

h) In My Goals he/she can check the goals that are assigned by the assignee, and even change his/her goal progress.

## **8. SOFTWARE REQUIREMENTS**

Technologies used are:

Python

MySQL

Python is platform independence. There are pros as well as cons with the interpreted languages. A high-level programming language class that executes many typical programming behaviors during runtime that static programming languages perform during compilation. Garbage collection alleviates the developer from manual memory control in which the programmer determines which artifacts to view and return to the memory system. It belongs to the class of high-level language. Garbage collection eliminates many bugs like. dangling pointers, double free bugs. It requires computational power when choosing which memory to spare. Python also support functional programming. Functional programming is a program framework where the code and composition functions are used to construct programs. It is a declarative programming model in which function descriptions are trees of expressions that each return a value, instead of a sequence of imperative statements that modify the state of the program of the environment. The basic library of a language is frequently regarded by its users as part of the language although it may have been viewed as a distinct body by the programmers. In addition to other parts that can be optionally implemented, several language standards describe a central collection that must be made accessible in all implementations. Therefore, the line between a language and its libraries varies from language to language. It has the standard library from which all the packages can be installed. There are many upgradations and new features available in the later versions of python. It can be installed in all different operating systems like Windows, macOS, Linux. Python is an open-source software programming language. The python software foundation manages the python software development and all its resources An operating system is machine software that administers computing tools and offers common computer programs services. Time-sharing operating systems plan activities for effective device use, which can include accounting tools for machine time management data storage, printing, and other services Python is supposed to be easy to read the language. Its formatting is physically uncluttered and uses English keywords while punctuation is used by other languages. In many other languages, the curly brackets are not used to delimit lines, and semicolons are available for sentences. It has fewer syntactic variants than C or Pascal and

special cases Instead of curly brackets or keywords, Python uses white space indentation to delimit the blocks. A rise in indentation occurs after certain statements, a reduction in indentation indicates the end of the current row.

## **8.1 PACKAGES USED**

### **Thinker:**

- This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit.
- Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application.
- Once created, these graphical elements can be associated with or interact with features, functionality, methods, data, or even other widgets.

### **Cosine\_similarity:**

- Cosine similarity is a measure used to measure how similar the documents are regardless of size.
- Mathematically, it measures the angle cosine in a multi-dimensional space between two vectors.
- The cosine similarity is useful because although the two identical documents are far apart from each other due to the Euclidean distance (due to the size of the document), they may still be closer to each other.
- The smaller the angle, the greater the similarity of the cosine.
- Cosine similarity is a metric used to determine how the documents are similar regardless of size.



## **JSP & JDBC:**

- Java Server Pages (JSP) it is a server-side technology which we used for creating web applications, and to create dynamic web content and we also used JDBC which acts as a bridge between middleware and backend databases (MySQL).

## **MySQL:**

- As a backend database for the web application we used MySQL database which is used for creating tables and storing data.
- It enables reducing the data flow between the user and server.
- It acts as a path for MySQL and python for data transfer.
- This package helps in datatype conversion from MySQL datatype to equivalent python datatype.
- It has its own library. It does not require any libraries from MySQL

## **9. EXPERIMENTAL SET UP AND RESULTS**

### **9.1 Dataset**

- We have collected the data from the website where the faculty updates the data and from the database which is already available in the MySQL server.

### **9.2 Procedure**

Step 1: User (Faculty) directly sign in with registered credentials in Guido chatbot.

Step 2: After validating registered credentials user (Faculty) will be redirected to the interface of the chatbot which displays three results, they are Notifications, Chat, and Task Findings.

Step 3: When faculty opens the Notifications tab then he/she is having different types of notifications that is pending tasks/goals, assigned tasks, change of Task Findings.

Step 4: When faculty opens the Chat tab then he/she can perform task updates, can change his/her goal progress, and manage their profile updates.

Step 5: When faculty opens the Task Findings tab then he/she can view his/her pending tasks and their deadlines, can update his task status using GuidoBot.

## 10. IMPLEMENTATION

### 10.1 Web application code:

#### Login page

```
<html>
<head>
  <script>
    if(!document.getElementById("us").checkValidity)
    {
      alert("Fill the username");
    }
    if(!document.getElementById("ps").checkValidity)
    {
      alert("Fill the username");
    }
  </script>
</head>
<body style="background-image: url(bg-01.jpg)">
<center>
  <form class="Main" action="logincheck.jsp" method="post">
    <h1 style="color: #065A7F "> Employee Login </h1>
    <div class="User">
      Username<br/>
      <input type="text" id="us" class="username" name="uname"
placeholder="Type your username" min="1" required>
    </div>
    <div class="Pass">
      Password<br/>
      <input type="password" name="pwd" id="ps" aria-
dropeffect="link" placeholder="Type the password" required>
    </div>
    <a href="#"><right>Forgot Password?</right></a>
    <br/>
    <input type="submit" value="Login" class="Login">
  </form>
```

</center>

</body>

</html>

**Home page:**

<html>

<head>

<title>Employee Home</title>

</head>

<body onload="myFunction()">

<center> <h1>A NextGen Website</h1></center>

<ul>

<li><a class="active" href="#home">Home</a></li>

<li><a href="profile.jsp">Profile </a></li>

<li><a href="AssignTask.jsp">Assign Task</a>

<li><a href="TaskTracking.jsp">Task Tracking </a></li>

<li><a href="AssignGoal.jsp" >Assign Goals</a></li>

<li><a href="GoalTracking.jsp" >Goal Tracking</a></li>

<li style="float:right"><a href="logout.jsp">Logout</a></li>

</ul>

</body>

</html>

**FacultyHomepage:**

<html>

<head>

<title>Employee Home</title>

</head>

<body>

<center> <h1>A NextGen Website</h1></center>

<ul>

<li><a class="active" href="#home">Home</a></li>

<li><a href="profile2.jsp">Profile </a></li>

<li><a href="MyTasks.jsp">My Tasks </a>

```
<li><a href="MyGoals.jsp">My Goals</a>
<li style="float:right"><a href="logout.jsp">Logout</a></li>
</ul>
</body>
</html>
```

**Profile page:**

```
<% @page import="java.sql.*"%>
<%
String uname=(String)session.getAttribute("uname");
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","srujan");
PreparedStatement pstmt = con.prepareStatement(" select * from emp_profile where
username=?");
pstmt.setString(1,uname);
ResultSet rs = pstmt.executeQuery();
while(rs.next())
{
%>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Profile</title>
</head>
<body>
<center> <h1>A NextGen Website</h1></center>
<ul>
<li><a href="EmpHome.jsp">Home</a></li>
<li><a class="active" href="#">Profile </a></li>
<li><a href="AssignTask.jsp">Assign Task</a></li>
<li><a href="TaskTracking.jsp">Task Tracking </a>
<li><a href="AssignGoal.jsp" >Assign Goals</a></li>
<li><a href="GoalTracking.jsp" >Goal Tracking</a></li>
```

```
<li style="float:right"><a href="logout.jsp">Logout</a></li>
</ul>
<div class="container">
  <form action="updateprofile.jsp">
    <div class="row">
      <div class="col-25">
        <label for="fname">Employee Id</label>
      </div>
      <div class="col-75">
        <input type="text" id="empid" readonly="readonly" name="id" placeholder="
<%=rs.getInt(1)%>">
      </div>
    </div>
    <div class="row">
      <div class="col-25">
        <label for="lname">Name</label>
      </div>
      <div class="col-75">
        <input type="text" id="fname" readonly="readonly" name="uname"
placeholder="<%=rs.getString(2)%>">
      </div>
    </div>
    <div class="row">
      <div class="col-25">
        <label for="subject">Email</label>
      </div>
      <div class="col-75">
        <input type="text" id="emailid" readonly="readonly" name="email"
placeholder="<%=rs.getString(4)%>">
      </div>
    </div>
    <div class="row">
      <div class="col-25">
        <label for="subject">Role</label>
      </div>
```

```
<div class="col-75">
  <input type="text" id="des" readonly="readonly" name="desg"
placeholder="<%=rs.getString(5)%>">
</div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Qualification</label>
  </div>
  <div class="col-75">
    <input type="text" id="qualy" readonly="readonly" name="qual"
placeholder="<%=rs.getString(6)%>">
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Research Publications</label>
  </div>
  <div class="col-75">
    <input type="text" id="reas" readonly="readonly" name="pub"
placeholder="<%=rs.getInt(7)%>">
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Experience</label>
  </div>
  <div class="col-75">
    <input type="text" id="em" readonly="readonly" name="exp"
placeholder="<%=rs.getInt(8)%>">
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="country">Department</label>
  </div>
```

[illegible]

### Update profile:

```
<% @page import="java.sql.*"%>
<%
String uname=(String)session.getAttribute("uname");
//int uid=Integer.parseInt(uname);
try {
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","srujan");
PreparedStatement pstmt = con.prepareStatement(" select * from emp_profile where
username=?");
pstmt.setString(1,uname);
ResultSet rs = pstmt.executeQuery();
while(rs.next())
{
```



```
%>
<html>
<head>
<title>Update Profile</title>
</head>
<body>
<center> <h1>A NextGen Website</h1></center>
<ul>
  <li><a href="EmpHome.jsp">Home</a></li>
  <li><a class="active" href="#">Profile </a></li>
  <li><a href="AssignTask.jsp">Assign Task </a></li>
  <li><a href="TaskTracking.jsp">Task Tracking </a>
  <li><a href="AssignGoal.jsp" >Assign Goals</a></li>
  <li><a href="GoalTracking.jsp" >Goal Tracking</a></li>
  <li style="float:right"><a href="logout.jsp">Logout</a></li>
</ul>
<div class="container">
  <form action="updateprofiledb.jsp">
    <div class="row">
      <div class="col-25">
        <label for="fname">Employee Id</label>
      </div>
      <div class="col-75">
        <input type="text" id="empid" readonly="readonly" name="id"
placeholder=" "<%=rs.getString(1)%>">
        <script>
          document.getElementById('empid').value="<%=rs.getString(1)%>" ;
        </script>
      </div>
    </div>
    <div class="row">
      <div class="col-25">
        <label for="lname">Name</label>
      </div>
      <div class="col-75">
```

```
<input type="text" id="fname" name="uname" placeholder="<%=rs.getString(2)%>">
<script>
document.getElementById('fname').value="<%=rs.getString(2)%>" ;
</script>
</div>
</div>
<div class="row">
<div class="col-25">
<label for="subject">Email</label>
</div>
<div class="col-75">
<input type="text" id="emailid" name="email" placeholder="<%=rs.getString(4)%>">
<script>
document.getElementById('emailid').value="<%=rs.getString(4)%>" ;
</script>
</div>
</div>
<div class="row">
<div class="col-25">
<label for="subject">Gender</label>
</div>
<div class="col-75">
<input type="radio" id="male" name="gender" value="male" >Male
<input type="radio" id="female" name="gender" value="female">Female
<script>
document.getElementById("<%=rs.getString(3)%>").checked=true ;
</script>
</div>
</div>
<div class="row">
<div class="col-25">
<label for="subject">Role</label>
</div>
<div class="col-75">
```

```
<select id="des" name="desg">
<option value="Faculty">Faculty</option>
<option value="Course Coordinator">Course Coordinator</option>
<option value="Dept. HoD">Dept. HoD</option>
<option value="HoD">HoD</option>
</select>
<script>
document.getElementById('des').value="<%=rs.getString(5)%>" ;
</script>
</div>
</div>
<div class="row">
<div class="col-25">
<label for="subject">Qualification</label>
</div>
<div class="col-75">
<select id="qal" name="qual">
<option value="M.Tech">M.Tech</option>
<option value="Ph. D">Ph.D</option>
<option value="B.Sc">B.Sc</option>
<option value="M.Sc">M.Sc</option>
</select>
<script>
document.getElementById('qal').value="<%=rs.getString(6)%>" ;
</script>
</div>
</div>
<div class="row">
<div class="col-25">
<label for="subject">Research Publications</label>
</div>
<div class="col-75">
<input type="text" id="reas" name="pub" placeholder="<%=rs.getInt(7)%>">
<script>
```

```
document.getElementById('reas').value="<%=rs.getInt(7)%>" ;
</script>
</div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Experience</label>
  </div>
  <div class="col-75">
    <input type="text" id="em" name="exp" placeholder="<%=rs.getInt(8)%>">
    <script>
      document.getElementById('em').value="<%=rs.getInt(8)%>" ;
    </script>
  </div>
</div>
<div class="row">
  <div class="col-25">
    <label for="country">Department</label>
  </div>
  <div class="col-75">
    <select id="de" name="dept">
      <option value="CSE">CSE</option>
      <option value="ECE">ECE</option>
      <option value="EEE">EEE</option>
      <option value="MECH">MECH</option>
      <option value="CIVIL">CIVIL</option>
      <option value="IT">IT</option>
    </select>
    <script>
      document.getElementById('de').value="<%=rs.getString(9)%>"
    </script>
  </div>
</div>
<div class="row">
```

[illegible]

```
</body>
</html>

<%
}
}
catch(Exception e)
{
}
%>
```

**Goal Tracking page:**

```
<% @page
import="java.sql.*,java.time.format.DateTimeFormatter,java.time.LocalDateTime,java.util.D
ate,java.text.SimpleDateFormat"%>

<%

String uname=(String)session.getAttribute("uname");

try {

    SimpleDateFormat myFormat = new SimpleDateFormat("yyyy-MM-DD");
    // String dateAfterString =;

    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd");

    LocalDateTime now = LocalDateTime.now();

    String dateAfterString=dtf.format(now);

    Class.forName("com.mysql.jdbc.Driver");

    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","srujan");

    PreparedStatement pstmt = con.prepareStatement(" select * from goal where assignerid=?");

    pstmt.setString(1,uname);

    ResultSet rs = pstmt.executeQuery();

    %>

<html>
```

```
<head>
<title>Update Profile</title>
<script>
function disableTxt( x,y){
    var val="save"+x;
    var btn="btn"+x;
    var prog="pro"+x;
    if(y=='pending')
        alert("Assignee still not Accepted this goal. Can't Update the progress")
    else if(y=='Rejected')
        alert("Assignee Rejected this goal to complete!!! can't update progress")
    else{
        document.getElementById(val).disabled=false;
        document.getElementById(prog).disabled=false;
        document.getElementById(btn).disabled=true;
    }
}

function view( x,y){
    if(y=='pending')
        alert("Assignee still not Accepted this goal.No Files are uploaded")
    else if(y=='Rejected')
        alert("Assignee Rejected this goal to complete!!! No files are Uploaded Yet.")
    else
        setTimeout(function(){ document.location.href='ViewFiles.jsp?gid='+x },400);
}

function Save(x){
    var v="id"+x;
    var s="pro"+x;
    var val=document.getElementById(v).value;
    var pro=document.getElementById(s).value;
    setTimeout(function(){ document.location.href='updateprogress.jsp?taskid='+val+'&pr
o='+ pro },0);
}
</script>
</head>
```

```
<body>
<center> <h1>A Guido Website</h1></center>

<%
int i=1;

PreparedStatement pstmt2 = con.prepareStatement(" select name from emp_profile where
id=?");

while(rs.next())
{

    pstmt2.setString(1,rs.getString(3));
    ResultSet rs2 = pstmt2.executeQuery();
    String name="";
    while(rs2.next())
    {
        name=rs2.getString(1);
    }
    %>

    <tr>

        <td id="tskid"><input disabled type="text" value="<%=rs.getInt(1) %>"
id="id<%=i%>"></td>

        <td><%= name%></td>
        <td><%= rs.getString(4)%></td>
        <td><%=rs.getString(6) %></td>
        <td id="dec"><%=rs.getString(7) %></td>

        <%

String dateBeforeString = rs.getString(5);
Date dateBefore = myFormat.parse(dateBeforeString);
Date dateAfter = myFormat.parse(dateAfterString);
long difference = dateAfter.getTime() - dateBefore.getTime();
int daysBetween =(int) (difference / (1000*60*60*24));
if(daysBetween<0)
{
    daysBetween=- (daysBetween);
    %>

    <td id="ded"><%=rs.getString(5) %><br>Days Remaining : <%=daysBetween %>
days</td>
```

```
<%
}
else
{
%>
    <td id="ded" style="color:red"><%=rs.getString(5) %><br>Overdue By :
<%=daysBetween %> days </td>
    <%
    }%>
    <%
String str=rs.getString(9);
    if(str.equalsIgnoreCase("Accepted"))
    {
        %>
        <td>Accepted</td>
        <%
    }
    else if(str.equalsIgnoreCase("Rejected"))
    {
%>
        <td>Rejected</td>
        <%
    }
    else if(str.equalsIgnoreCase("pending"))
    {
%>
        <td>Pending</td>
        <%
    }
    %>
    <td><input id="pro<%=i %>" disabled type="text" value="<%=
rs.getString(8)%>"></td>
    <td id="chng">
        <button onclick="disableTxt(<%=i %>,<%=rs.getString(9)%>)" id="btn<%=i
%>">Update</button>
```



```
<button disabled onclick="Save(<%=i %>)" id="save<%=i%>">Save</button>
</td>
<td><button onclick="view('<%=rs.getString(1)%>','<%=rs.getString(9)%>')"
>View</button></td>
</tr>
<%
i++;
}
%>
</table>
</form>
</div>
<script>
</script>
</body>
</html>
<%
}
catch(Exception e)
{
}
%>
```

**Task Tracking page:**

```
<% @page
import="java.sql.*,java.time.format.DateTimeFormatter,java.time.LocalDateTime,java.util.Date,java.text.SimpleDateFormat"%>
<%
//String uname="170030875";
String uname=(String)session.getAttribute("uname");
//int uid=Integer.parseInt(uname);
try {
    SimpleDateFormat myFormat = new SimpleDateFormat("yyyy-MM-DD");
    // String dateAfterString =;
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd");
```

```
LocalDateTime now = LocalDateTime.now();
String dateAfterString=dtf.format(now);
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test","root","srujan");
PreparedStatement pstmt = con.prepareStatement(" select * from task where assignerid=?
&& acceptance=?");
pstmt.setString(2,"Pending");
pstmt.setString(1,uname);
ResultSet rs = pstmt.executeQuery();
%>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Update Profile</title>
<script>
function Accept(x){
    var v="id"+x;
    var s="stat"+x;
    var comment="";
    var val=document.getElementById(v).value;
    var stat="Approved";
    var comm=document.getElementById(val)
    if(comm!=null)
        comment=comm.value;
    else
        comment="Good Work"

    setTimeout(function(){ document.location.href='updatestatus.jsp?taskid='+val+'&stat=
'+stat+'&comm='+comment },400);
}
function Reject(x){
    var v="id"+x;
    var s="stat"+x;
    var c="com"+x;
    var val=document.getElementById(v).value;
```

```
var comment=document.getElementById(c).value;
var stat="Rejected";
if(comment.length<1)
    alert("Mention the reson for Rejection")
else
    setTimeout(function(){ document.location.href="updatestatus.jsp?taskid="+val+"&stat="+stat+"&comm="+comment },500);
}
</script>
</head>
<body>
<center> <h1>A Guido Website</h1></center>
<ul>
<li><a href="EmpHome.jsp">Home</a></li>
<li><a href="profile.jsp">Profile </a></li>
<li><a href="AssignTask.jsp">Assign Task </a></li>
<li><a class="active" href="#">Task Tracking </a>
<li><a href="AssignGoal.jsp" >Assign Goals</a></li>
<li><a href="GoalTracking.jsp" >Goal Tracking</a></li>
<li style="float:right"><a href="logout.jsp">Logout</a></li>
</ul>
<div class="container">
<form action="#">
<table id="tasks" >
<tr>
<th><h4>TaskId</h4></th>
<th><h4>Assigned To</h4></th>
<th><h4>Priority</h4></th>
<th><h4>Type</h4></th>
<th><h4>Description</h4></th>
<th><h4>DeadLine</h4></th>
<th><h4>Status</h4></th>
<th><h4>Acceptence</h4></th>
<th><h4>Comments</h4></th>
</tr>
```

```
<%
int i=1;
PreparedStatement pstmt2 = con.prepareStatement(" select name from emp_profile where
id=?");
while(rs.next())
{
    pstmt2.setString(1,rs.getString(3));
    ResultSet rs2 = pstmt2.executeQuery();
    String name="";
while(rs2.next())
    {
        name=rs2.getString(1);
    }
    %>
    <tr>
        <td id="tskid"><input disabled type="text" value="<%=rs.getInt(1) %>"
id="id<%=i%>"></td>
        <td><%= name%></td>
        <td id="type"><%= rs.getString(4)%></td>
        <td id="type2"><%=rs.getString(6) %></td>
        <td id="dec"><%=rs.getString(7) %></td>
    <%
        String dateBeforeString = rs.getString(5);
        Date dateBefore = myFormat.parse(dateBeforeString);
        Date dateAfter = myFormat.parse(dateAfterString);
        long difference = dateAfter.getTime() - dateBefore.getTime();
        int daysBetween =(int) (difference / (1000*60*60*24));
        if(daysBetween<0)
        {
            daysBetween=- (daysBetween);
        }
        %>
        <td id="ded"><%=rs.getString(5) %><br>Days Remaining : <%=daysBetween %>
days</td>
    <%
    }
}
```

```
        else
        {
            %>
            <td id="ded" style="color:red"><%=rs.getString(5) %><br>Overdue By :
<%=daysBetween %> days </td>
            <%
            }%>
            <td><%= rs.getString(8)%></td>
            <td>
            <button onclick="Accept(<%=i %>)" >Approve</button>
            <button onclick="Reject(<%=i %>)" >Reject</button>
            </td>
            <td><input id="com<%=i%>" type="text"></td>
            </tr>
            <%
            i++;
        }
        %>
        </table>
        </form>
    </div>
    <script>
    </script>
    </body>
    </html>
    <%
    }
    catch(Exception e)
    {
    }
    %>
```

## **10.2 GuidoBot:**

**Login page :**

```
from tkinter import *
import os
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root"
)
global mycursor
mycursor = mydb.cursor()
# Designing window for login
def login():
    global login_screen
    login_screen = Toplevel(main_screen)
    login_screen.title("Login")
    login_screen.geometry("300x250")
    Label(login_screen, text="Please enter details below to login").pack()
    Label(login_screen, text="").pack()
    global username_verify
    global password_verify
    username_verify = StringVar()
    password_verify = StringVar()
    global username_login_entry
    global password_login_entry
    Label(login_screen, text="Username * ").pack()
    username_login_entry = Entry(login_screen, textvariable=username_verify)
    username_login_entry.pack()
    Label(login_screen, text="").pack()
    Label(login_screen, text="Password * ").pack()
    password_login_entry = Entry(login_screen, textvariable=password_verify, show= '*')
    password_login_entry.pack()
    Label(login_screen, text="").pack()
    Button(login_screen, text="Login", width=10, height=1, command = login_verify).pack()
def login_verify():
```

```
username1 = username_verify.get()
password1 = password_verify.get()
query="select * from test.emp_profile where username= %s and password=%s"
val=(username1,password1)
mycursor.execute(query,val)
result=mycursor.fetchone()
if result is not None:
    login_sucess(username1)
else:
    user_not_found()
def login_sucess(username1):
    login_screen.destroy()
    script_descriptor = open("./LandingPage.py")
    a_script = script_descriptor.read()
    sys.argv = [username1]
    exec(a_script)
def user_not_found():
    global user_not_found_screen
    user_not_found_screen = Toplevel(login_screen)
    user_not_found_screen.title("Success")
    user_not_found_screen.geometry("150x100")
    Label(user_not_found_screen, text="User Not Found").pack()
    Button(user_not_found_screen, text="OK",
command=delete_user_not_found_screen).pack()
def delete_user_not_found_screen():
    user_not_found_screen.destroy()
# Designing Main(first) window
def main_account_screen():
    global main_screen
    main_screen = Tk()
    main_screen.geometry("250x250")
    main_screen.title("Account Login")
    Label(text="Login to your Account ", bg="blue", width="300", height="2",
font=("Calibri", 13)).pack()
    Label(text="").pack()
```

```
Button(text="Login", height="2", width="30", command = login).pack()
main_screen.mainloop()
main_account_screen()
Landing Page:
from tkinter import *
import sys
import os
from tkinter import *
import time
import tkinter.messagebox

import math
import re
from collections import Counter
#from csvgui import response
import threading
import mysql.connector
import mysql.connector
global id
global name
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="srujan"
)
global mycursor
mycursor = mydb.cursor()
id=sys.argv[0]
def open_window():
    script_descriptor = open("./Chatpage.py")
    ab_script = script_descriptor.read()
    sys.argv = [id]
    exec(ab_script,locals(),locals())
def notification():
```



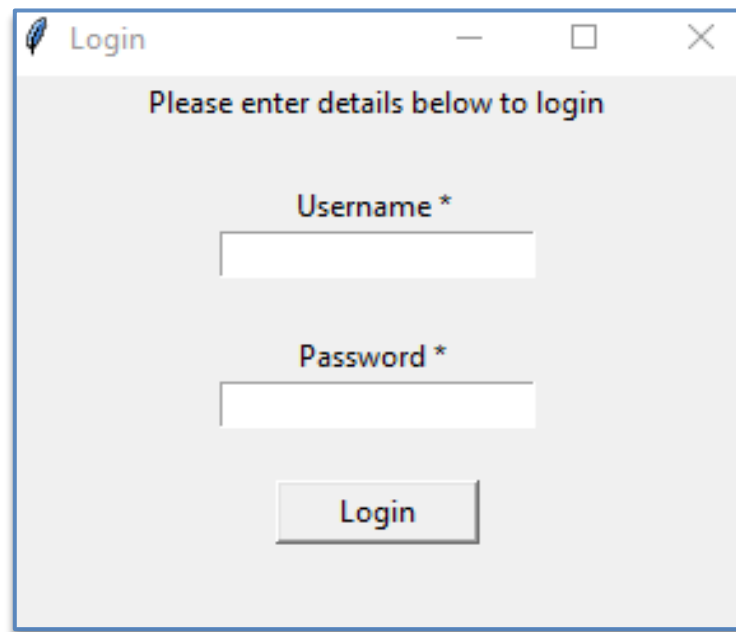
```
script_descriptor = open("./Notificationspage.py")
ab_script = script_descriptor.read()
sys.argv = [id]
exec(ab_script)
root = Tk()
button1 = Button(root, text="Notification", command=notification)
button2 = Button(root, text="Chat", command=open_window)
button1.pack()
button2.pack()
root.title("Home")
root.geometry("300x300+120+120")
root.mainloop()
Notifications Page:
import tkinter as tk
import sys
from tkinter import *
import time
import tkinter.messagebox
import math
import re
from collections import Counter
#from csvgui import response
import threading
import mysql.connector
global id
global name
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="srujan"
)
global mycursor
mycursor = mydb.cursor()
id=sys.argv[0]
```

```
saved_username = ["You"]
window_size="690x420"
text = "
class ChatInterface(Frame):
    def init(self, master=None):
        global text
        Frame.init(self, master)
        self.master = master
        self.tl_bg = "black"
        self.tl_bg2 = "black"
        self.tl_fg = "black"
        self.font = "Verdana 10"
        menu = Menu(self.master)
        self.master.config(menu=menu, bd=5,bg="black")
        self.text_frame = Frame(self.master, bd=6,bg="black")
        self.text_frame.pack(expand=True, fill=BOTH)
        self.text_box_scrollbar = Scrollbar(self.text_frame, bd=0)
        self.text_box_scrollbar.pack(fill=Y, side=RIGHT)
        self.text_box = Text(self.text_frame, yscrollcommand=self.text_box_scrollbar.set,
state=DISABLED,
                        bd=1, padx=4, pady=4, spacing3=8, wrap=WORD, bg="black",
font="Verdana 10", relief=GROOVE,
                        width=10, height=1,fg="white")
        self.text_box.pack(expand=True, fill=BOTH)
        self.text_box_scrollbar.config(command=self.text_box.yview)
        self.entry_frame = Frame(self.master, bd=1,bg="black")
        self.entry_frame.pack(side=LEFT, fill=BOTH, expand=True)
        self.entry_field = Entry(self.entry_frame, bd=1, justify=LEFT)
        self.entry_field.pack(fill=X, padx=4, pady=4, ipady=3)
        self.send_button_frame = Frame(self.master, bd=0)
        self.send_button_frame.pack(fill=None)
        self.send_button = Button(self.send_button_frame, text="Text", width=5,
relief=GROOVE, bg='white',
                        bd=1, command=lambda: self.send_message_insert(1,None),
activebackground="#FFFFFF",
                        activeforeground="#000000")
```

```
self.send_button_frame2 = Frame(self.master, bd=0)
self.send_button_frame2.pack(fill=BOTH)
self.send_button2 = Button(self.send_button_frame2, text='voice', width=5,
relief=GROOVE, bg='white',
                        bd=2, command=lambda: self.send_message_insert(2,None),
activebackground="#FFFFFF",
                        activeforeground="#000000")
#self.send_button2.config(image=icon1)
self.send_button.pack(side=LEFT, ipady=8)
self.send_button2.pack(side=LEFT, ipady=8)
self.master.bind("<Return>", self.send_message_insert)
def chatexit(self):
    exit()
root=Tk()
txt=Text(root)
txt.grid(row=0,column=0,columnspan=1)
query2="select name from test.emp_profile where id=%s"
val2=(str(id),)
mycursor.execute(query2,val2)
res=mycursor.fetchone()
name=res[0]
string1=""Hi, { name} \n
        Here are the pending tasks of your's
        """.format(name=name)
txt.insert(END, "\n"+string1)
query="select * from test.task where assignedTo=%s and status=%s"
val=(id,"assigned")
mycursor.execute(query,val)
result=mycursor.fetchall()
i=1
for x in result:
    query1="select name from test.emp_profile where id=%s"
    val1=(x[1],)
    mycursor.execute(query1,val1)
    result1=mycursor.fetchone()
```

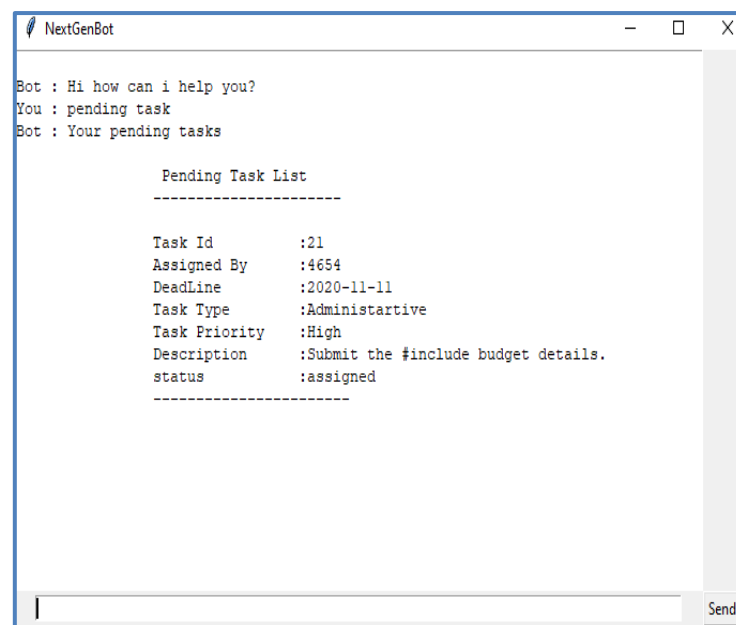
```
query=""\n    Pending Task ""+str(i)+"""\n\n    -----\n\n    Assigned By :{assign}\n    DeadLine   :{deadline}\n    Task Type   :{types}\n    Task Priority :{priority}\n    Description :{desc}\n\n    *****\n\n    """.format(assign=x[1],deadline=x[4],types=x[5],priority=x[3],desc=x[6])\n\n    i=i+1\n\n    txt.insert(END, "\\n"+query)\n\na = ChatInterface(root)\nroot.geometry(window_size)\nroot.title("Notifications")\n#root.iconbitmap('favicon.ico')\nroot.mainloop()
```

## **11. OUTPUT SCREENS**



A screenshot of a login window titled "Login". The window has a light gray background and a blue border. It contains the text "Please enter details below to login" at the top. Below this, there are two input fields: "Username \*" and "Password \*". At the bottom, there is a "Login" button.

Fig 5: The above figure depicts that User need to enter his/her valid username and password to login to the Guido Bot Interface.



A screenshot of a chat window titled "NextGenBot". The window has a white background and a blue border. It shows a conversation between the bot and the user. The bot says "Hi how can i help you?". The user says "pending task". The bot responds "Your pending tasks". Below this, there is a "Pending Task List" section with a table of task details. At the bottom, there is a text input field and a "Send" button.

Pending Task List	
Task Id	:21
Assigned By	:4654
DeadLine	:2020-11-11
Task Type	:Administrative
Task Priority	:High
Description	:Submit the #include budget details.
status	:assigned

Fig 6: The above figure represents that Chatbot displays the pending tasks of the user.

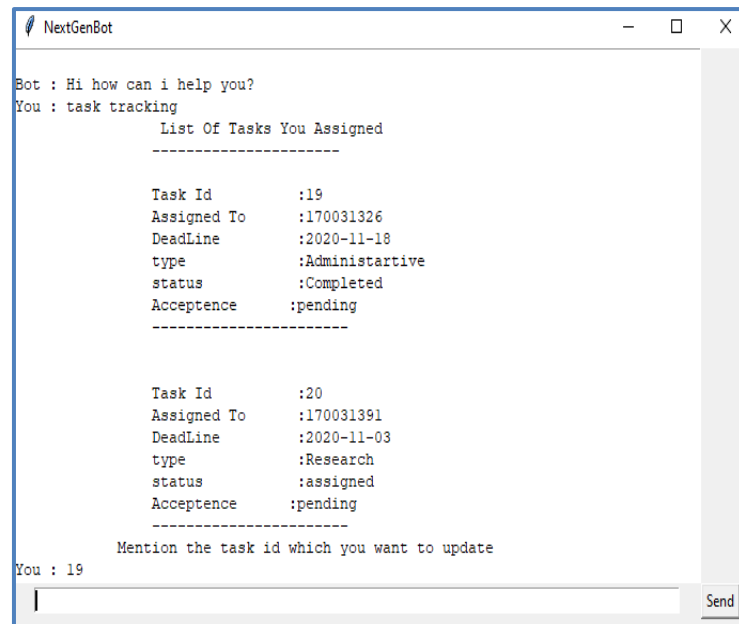


Fig 7: The above figure reveals that assigner can track the task that he assigned, initially chatbot displays all the tasks that are assigned by him.

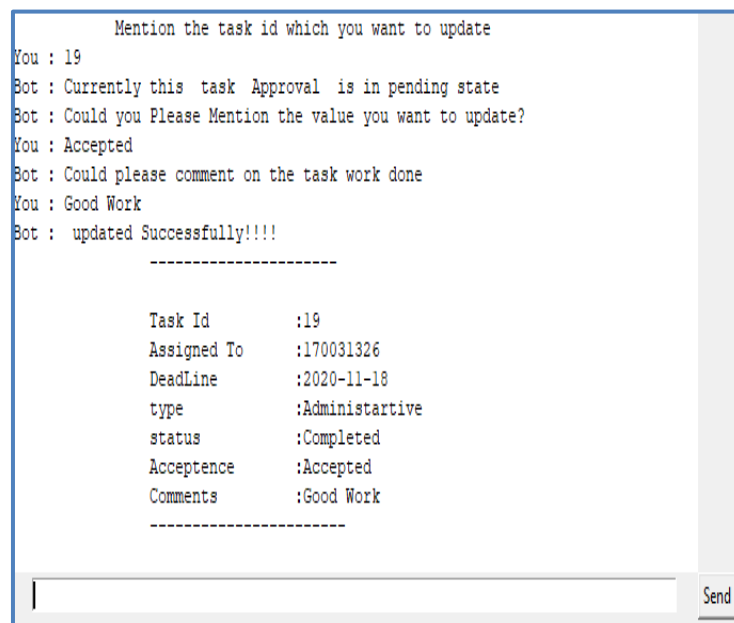


Fig 8: The above figure represents that the assigner can track the task by mentioning the task id and he can accept or reject the task. User can also mention his comments regarding the work.

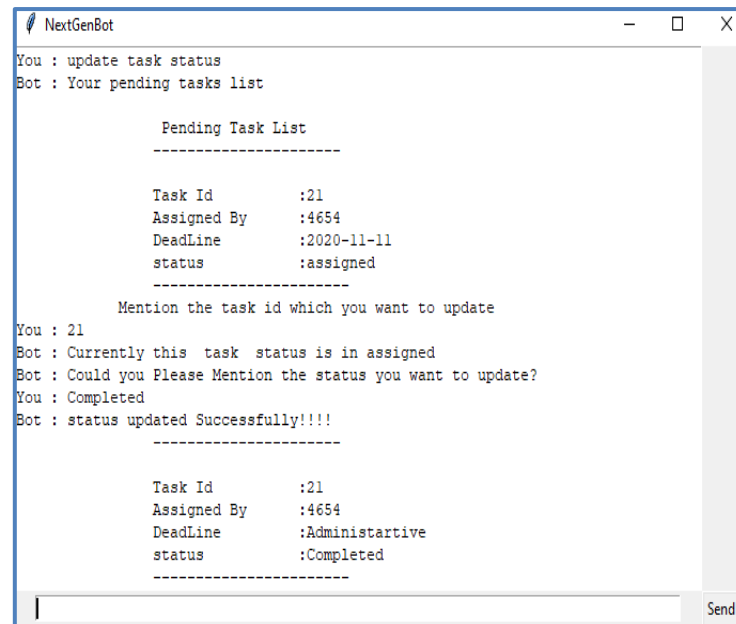


Fig 9: The above figure initially displays all the tasks that are assigned to the assignee and by mentioning the task id assignee can update the status (InProgress / Completed) of his task.

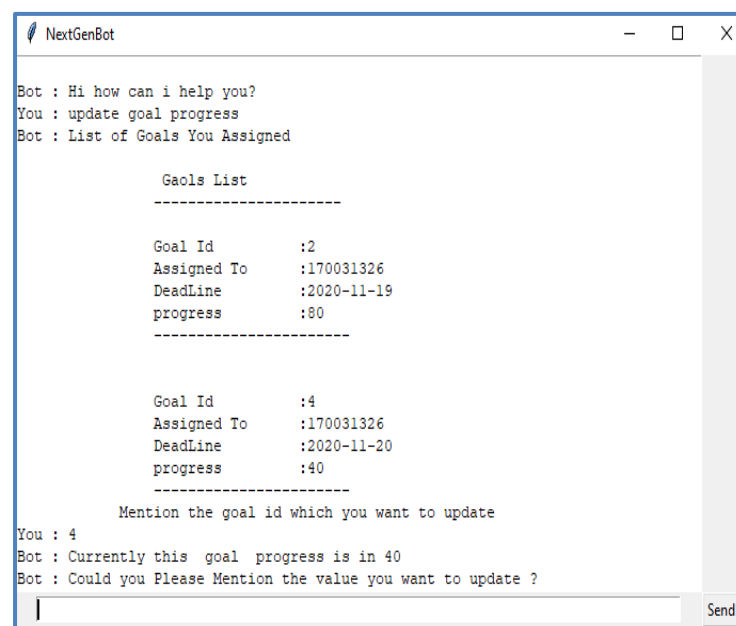


Fig 10: The above figure represents how to interact with the chatbot to update the goal progress. Chatbot displays the list of goals assigned by the assigner. With the help of Goal id assigner can update the goal progress.

Mention the goal id which you want to update

You : 4

Bot : Currently this goal progress is in 40

Bot : Could you Please Mention the value you want to update ?

You : 50

Bot : Progress updated Successfully!!!!

-----

Task Id	:4
Assigned To	:170031326
DeadLine	:Academic
Progress	:50

-----

Fig 11: The above figure represents the initial progress of the goal in the chatbot. Assigner can increment it with the percentage of completion as a progress.

**A NextGen Website**

Home Profile Assign Task Task Tracking **Assign Goals** Goal Tracking Logout

**Assign Goals**

Assign To

Type

Priority

DeadLine

Description

Fig 12: The above figure displays a webpage where Assigner can assign goals to Assignee by stating its priority and deadline to complete it.



A NextGen Website									
Home		Profile		My Tasks		My Goals		Logout	
Goalid	Assigned By	Priority	Type	Description	DeadLine	Progress(%)	Acceptance status	Upload Files	
1	Srujan	Medium	Research	complete paper publication	2020-11-27 Days Remaining : 14 days	0	<input type="button" value="Accept"/> <input type="button" value="Reject"/>	<input type="button" value="Upload"/>	
2	Srujan	Medium	Academic	Complete RPA Certifications	2020-11-19 Days Remaining : 6 days	80	<input type="button" value="Accepted"/>	<input type="button" value="Upload"/>	
3	Srujan	Medium	Academic	Complete Coursera courses in DS	2020-11-27 Days Remaining : 14 days	0	<input type="button" value="Rejected"/>	<input type="button" value="Upload"/>	
4	Srujan	Medium	Academic	Complete Edx Courses	2020-11-20 Days Remaining : 7 days	40	<input type="button" value="Accepted"/>	<input type="button" value="Upload"/>	

Fig 13: The above picture represents My Goals tab of Assignee, where he can view the goals that are assigned to him and he accept/reject the goal, checks its progress and able to upload the work that he completed in order to increase the progress.

A NextGen Website									
Home		Profile		Assign Task		Task Tracking		Goal Tracking	
Goalid	Assigned To	Priority	Type	Description	DeadLine	Status	Progress(%)	Update progress View Files	
1	Maresh	Medium	Research	complete paper publication	2020-11-27 Days Remaining : 14 days	Pending	0	<input type="button" value="Update"/> <input type="button" value="Save"/>	<input type="button" value="View"/>
2	Maresh	Medium	Academic	Complete RPA Certifications	2020-11-19 Days Remaining : 6 days	Accepted	80	<input type="button" value="Update"/> <input type="button" value="Save"/>	<input type="button" value="View"/>
3	Maresh	Medium	Academic	Complete Coursera courses in DS	2020-11-27 Days Remaining : 14 days	Rejected	0	<input type="button" value="Update"/> <input type="button" value="Save"/>	<input type="button" value="View"/>
4	Maresh	Medium	Academic	Complete Edx Courses	2020-11-20 Days Remaining : 7 days	Accepted	40	<input type="button" value="Update"/> <input type="button" value="Save"/>	<input type="button" value="View"/>

Fig 14: The above picture represents Goal Tracking page of Assigner, where he/she can view complete data along with deadline and status of goals that he/she assigned and he/she can be able to change the progress of goals based on work done by assignee.

A NextGen Website									
Home		Profile		My Tasks		My Goals		Logout	
TaskId	Assigned By	Priority	Type	Description	DeadLine	Status	Modify	Acceptance	
19	Srujan	Medium	Administrative	Submit the CRT Students placement status	2020-11-18 Days Remaining : 5 days	Completed	<a href="#">Change Status</a> <a href="#">Save</a>	Submitted for Approval	
22	Surya Kiran	Medium	Academics	submit test 2 marks along with list of students got less than 40%.	2020-11-21 Days Remaining : 8 days	In Progress	<a href="#">Change Status</a> <a href="#">Save</a>	Yet To be completed	
25	Srujan	High	Academics	Submit test 2 Results	2020-11-01 Overdue By : 12 days	Assigned	<a href="#">Change Status</a> <a href="#">Save</a>	Rejected *Not Submitted InTime	

Fig 15: The above figure represents the My Tasks page of Assignee, where he can view all the tasks that are assigned to him and he can be able to change their status based on his work done and can view final Acceptance of his work from Assigner.

A NextGen Website									
Home		Profile		Assign Task		Task Tracking		Assign Goals	
								Goal Tracking	
TaskId	Assigned To	Priority	Type	Description	DeadLine	Status	Acceptance	Comments	
19	Mahesh	Medium	Administrative	Submit the CRT Students placement status	2020-11-18 Days Remaining : 5 days	Completed	<a href="#">Approve</a> <a href="#">Reject</a>		
20	Poojitha	High	Research	Complete 3 the Paper Publications in this semester in the field of AI.	2020-11-03 Overdue By : 10 days	assigned	<a href="#">Approve</a> <a href="#">Reject</a>		

Fig 16: The above picture represents the Task Tracking page of Assigner where he can track all the tasks that he assigned along with their deadlines and able to check their status and Approves the tasks.

## **12. CONCLUSION**

Technology as a concept is continuously working in providing new ways to save our time, and effort. In this research we are providing a chatbot assistance for the application Faculty Goal Tracking system. The Guido bot is like having a personal assistant that exists in your phone or computer. Instead of someone else running your schedule user can do it on his own. Faculty can utilize this Guido bot to update their profile, manage their goals and tasks in the goal tracking application in the best possible manner with the timely updates and notifications.

### **13. REFERENCES**

- [1] Sreelakshmi, A., Abhinaya, S., Nair, A., & Jaya Nirmala, S. (2019). A question answering and quiz generation chatbot for education. 2019 Grace Hopper Celebration India (GHCI).
- [2] Shinde, M. S., Nahar, M. S., & Shelke, S. (2019). Intelligent chatbot for guided navigation of repository contend. IJARCCCE, 8(5).
- [3] Sandu, N., & Gide, E. (2019). Adoption of AIchabots to enhance student learning experience in higher education in India. 2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET)
- [4] Selvi, V., Saranya, S., Chidida, K., & Abarna, R. (2019). Chatbot and bullyfree chat. 2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN).
- [5] Sivcevic, D., Kosanin, I., Nedeljkovic, S., Nikolic, V., Kuk, K., & Nogo, S. (2020). Possibilities of used intelligence based agents in instant messaging on egovernment services. 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH).
- [6] Pandian, A. P. (2019). ARTIFICIAL INTELLIGENCE APPLICATION IN SMART WAREHOUSING ENVIRONMENT FOR AUTOMATED LOGISTICS. Journal of Artificial Intelligence, 1(02), 63- 72.
- [7] Niranjana, P. Y., Rajpurohit, V. S., & Malgi, R. (2019). A survey on chat-bot system for agriculture domain. 2019 1st International Conference on Advances in Information Technology (ICAIT).
- [8] Oh, K., Lee, D., Ko, B., & Choi, H. (2017). A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. 2017 18th IEEE International Conference on Mobile Data Management (MDM).
- [9] Punjabi, S., Scthuram, V., Ramachandran, V., Boddu, R., & Ravi, S. (2019). Chat bot using API : Human to machine conversation. 2019 Global Conference for Advancement in Technology (GCAT).
- [10] Chandy, A. (2019). A REVIEW ON IOT BASED MEDICAL IMAGING TECHNOLOGY FOR HEALTHCARE APPLICATIONS. Journal of Innovative Image Processing (JIIP), 1(01), 51-60.

- [11] Akhila, G., Madhubhavana, N., Ramareddy, N. V., Hurshitha, M., & Ravinder, N. (2018). A survey on health prediction using human activity patterns through smart devices. *International Journal of Engineering and Technology(UAE)*, 7(1), 226-229.
- [12] Balram, G., & Kiran Kumar, K. (2018). Smart farming: Disease detection in crops. *International Journal of Engineering and Technology(UAE)*, 7(2), 33-36.
- [13] Buvaneshvari, M., Uthayakumar, J., & Amudhavel, J. (2017). Fuzzy based clustering to maximize network lifetime in wireless mobile sensor networks. *Journal of Advanced Research in Dynamical and Control Systems*, 9(Special Issue 12), 2133-2143.
- [14] Umar, S., Subbarayudu, Y., Kumar, K. K., & Bashwanth, N. (2017). Designing of dynamic reclustering leach protocol for calculating total residual time and performance. *International Journal of Electrical and Computer Engineering*, 7(3), 1286- 1292.
- [15] Vijay Bhaskar Reddy, Y., Reddy, L. S. S., & Sai Satya Naryana Reddy, S. (2017). Comparative study of density-based clustering algorithms. *International Journal of Civil Engineering and Technology*, 8(12), 763-767.
- [16] Kanaka Durga, K., & Rama Krishna, V. (2016). Automatic detection of illegitimate websites with mutual clustering. *International Journal of Electrical and Computer Engineering*, 6(3), 995-1001.
- [17] Kousar Nikhath, A., & Subrahmanyam, K. (2016). Incremental evolutionary genetic algorithm based optimal document clustering (ODC). *Journal of Theoretical and Applied Information Technology*, 87(3), 479-488.
- [18] Nirmal, K. R., & Satyanarayana, K. V. V. (2016). Issues of K means clustering while migrating to map reduce paradigm with big data: A survey. *International Journal of Electrical and Computer Engineering*, 6(6), 3047-3051.
- [19] Durga Rajesh, K. V., Krishna Akhil, G., Teja Kumar, A., Vivek, K., & Radha Krishna, B. (2017). An effective similarity based sheep flock heredity algorithm to anticipate number of cells. *Journal of Advanced Research in Dynamical and Control Systems*, 9(Special issue 14), 2720-2726.
- [20] Raman, D., Bezawada, B., Rajinikanth, T. V., & Sathyanarayan, S. (2017). Static program behavior tracing for program similarity quantification

# Paper 2\_2.0

## ORIGINALITY REPORT

# 5%

### SIMILARITY INDEX

#### PRIMARY SOURCES

- |   |  |                 |
|---|--|-----------------|
| 1 | <a href="http://www.chatcompose.com">www.chatcompose.com</a><br>Internet   | 48 words — 1%   |
| 2 | Kyo-Joong Oh, Dongkun Lee, Byungsoo Ko, Ho-Jin Choi. "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation", 2017 18th IEEE International Conference on Mobile Data Management (MDM), 2017<br>Crossref | 27 words — 1%   |
| 3 | <a href="http://www.ijstr.org">www.ijstr.org</a><br>Internet   | 27 words — 1%   |
| 4 | <a href="http://www.semanticscholar.org">www.semanticscholar.org</a><br>Internet   | 20 words — < 1% |
| 5 | Dusko Sivcevic, Ivan Kosanin, Slobodan Nedeljkovic, Vojkan Nikolic, Kristijan Kuk, Srdan Nogo. "Possibilities of used intelligence based agents in instant messaging on e-government services", 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), 2020<br>Crossref   | 19 words — < 1% |
| 6 | <a href="http://hdl.handle.net">hdl.handle.net</a><br>Internet   | 19 words — < 1% |
| 7 | <a href="http://researchr.org">researchr.org</a><br>Internet   | 19 words — < 1% |
| 8 | Prashant Y. Niranjana, Vijay S. Rajpurohit, Rasika Malgi. "A Survey on Chat-Bot system for Agriculture   | 17 words — < 1% |

---

9 "GCAT 2019 Papers", 2019 Global Conference for Advancement in Technology (GCAT), 2019 15 words — < 1%  
Crossref

---

10 Shreya Karve, Arati Nagmal, Sahil Papalkar, S. A. Deshpande. "Context Sensitive Conversational Agent Using DNN", 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018 9 words — < 1%  
Crossref

---

11 psasir.upm.edu.my 8 words — < 1%  
Internet

---

EXCLUDE QUOTES OFF  
EXCLUDE BIBLIOGRAPHY ON

EXCLUDE MATCHES OFF



# 6<sup>th</sup> International Conference on Inventive Computation Technologies

20-22 January, 2021 | Coimbatore, India

[icicts.com/2021/](http://icicts.com/2021/) | [icict.conf@gmail.com](mailto:icict.conf@gmail.com)

## ACCEPTANCE LETTER

To:

Nagothu. Srujan, Utlapalli. Mahesh, Vemuri. Poojitha,  
Syam. Jason, Jonnalagadda. Surya Kiran,  
Department of Computer Science and Engineering,  
Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., India-522502.

Herewith, the conference committee is happy to inform you that the peer reviewed **Paper Id: ICICT-0101** entitled **“GuidoBot : A Novel Approach for Assisting Faculty in Tracking Goals and finding tasks”** has been accepted for presentation as well as inclusion in the conference proceedings of the **ICICT 2021: 6th International Conference on Inventive Computation Technologies (ICICT-2021)** to be held in Coimbatore, Tamilnadu, India during **20-22 January, 2021**.

We look forward to your participation in the ICICT 2021

All registered and presented papers will be published in Accepted papers will be submitted for inclusion into IEEE Xplore.

Sincerely,

Dr.S. Smys,  
Professor,  
Department of CSE,  
RVS Technical Campus,  
Coimbatore,  
India.

Previous Publications

[ICICT 2020](#)  
[ICECA 2019](#)  
[ICICT 2018](#)  
[ICECA 2017](#)  
[ICICT 2016](#)

Proceedings by

