**CHAT ASSISTANT**

*A*

*Mini Project Report*

*Submitted in partial fulfillment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**NARSINI SHRAVANI(1602-20-737-039)**

**SINGARI TEJASWI(1602-20-737-055)**

**RAMAVATH KARTHIK(1602-20-737-018)**



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University and Approved by AICTE)**

**Ibrahimbagh, Hyderabad-31, 2022**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE**

**(Affiliated to Osmania University and Approved by AICTE)**

**Hyderabad - 500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

We, **NARSINI SHRAVANI,** **SINGARI** **TEJASWI**, and **RAMAVATH KARTHIK,** bearing hall ticket numbers, **1602-20-737-039, 1602-20-737-055** and **1602-20-737-018**, hereby declare that the project report entitled **”CHAT ASSISTANT”** is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**~Narsini Shravani(1602-20-737-039)**

**~ Singari Tejaswi (1602-20-737-055)**

**~ Karthik Ramavath(1602-20-737-018)**

(Faculty In-Charge) (Head,Dept of IT)

**ACKNOWLEDGMENT**

We extend our sincere thanks to Dr. S. V. Ramana, Principal, Vasavi College of Engineering for his encouragement.

We express our sincere gratitude to Dr. K. Ram Mohan Rao, Professor & Head, Department of Information Technology, Vasavi College of Engineering, for introducing the Mini-Project module in our curriculum, and also for his suggestions, motivation, and co-operation for the successful completion of our Mini Project.

We also want to thank and convey our gratitude towards our mini project coordinators DIVYA LINGANENI and RAJYALAKSHMI, for guiding us in understanding the process of project development & giving us timely suggestions at every phase.

We would also like to sincerely thank the project reviewers for their valuable inputs and suggestions.

**ABSTRACT**

The main objective of the project is to propose a concept for having an app or website or like this chat assistant for colleges. This project is useful for any kind of person, especially students and their parents who want to know about the college details in detail.

When a person wants to get any kind of information related to college by using this chat assistant, firstly, they have to register by creating an account with username and password. Then, they can login to the vasavi chat assistant with these credentials, so that they can clarify or get their answers by asking information related to the college like faculty, books availability in library, about departments, about college websites etc.

If a user already creates an account for one time then there is no need to create an account for the next time he can use the created account, since it stores permanently.

The framework which was used in this project was GUI using tkinter in python.

**TABLE OF CONTENTS**

**1.Introduction**

a. Information

b. Features

**2.Technology**

a. Software Requirements

b. Hardware Requirements

**3.Proposed Work**

a. Design

b. Implementation

c. Testing

**4.Results**

**5.Outcomes apart from curriculum**

**6.Conclusion and future work**

**7. References**

**INTRODUCTION**

**a. Information**

Every student or their parents must have anxiousness to know about the college while joining the college or taking admission. At that time, they have to wait upto going to college and experience it directly. Instead of this, at this time, college websites or apps or chat assistants or chatbots are mainly useful to know about college information even if we don’t get 100% of idea on it but it can help us to give at least to 50% of information of our need.

Even now most of the colleges are providing online classes instead of offline classes due to the pandemic. By this most of the students are behind to know the information about college and announcements so a website or an app or a chat assistant will help the students based on their requirement what they need to know.

Even in these, a chat assistant gives a straightforward answer to the enquired question based on its database. But in any website or app we have to search for it which consumes a student’s time.

**b. Features**

Our main objective is to create a “Chat Assistant” which is able to give all the required information about vasavi college.

In this, we can get information about College details like all its departments.

* HOD’s and core subjects for each department.
* Library details and books availability.
* College website and exam platform website.
* Contact information for each sector.
* About transportation.

**Technology**

**a. Software Requirements**

* We use the pycharm community in order to develop the code for the chat assistant.
* We use files in order to store the data for registration i.e. username and password with the username as file name.
* Operating System. We have chosen the Windows operating system for its best support and user friendliness.

**b. Hardware Requirements**

The hardware components used in order to create the chat assistant are as follows:

* Computer with 2GB RAM
* Internet
* External drives for backup

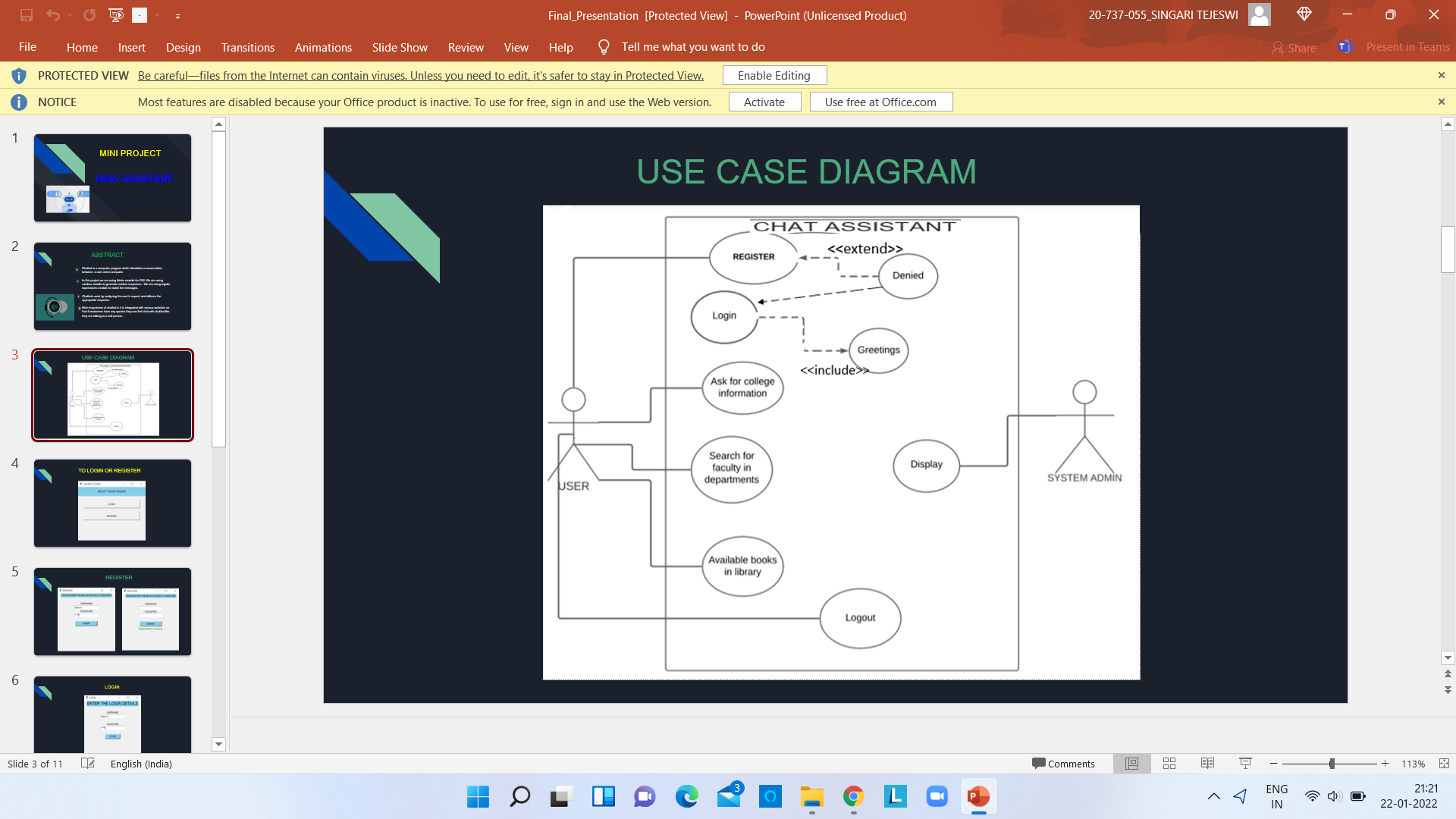
**PROPOSED WORK**

**a. Design**

**Use Case Diagram**

A UML use case diagram is the primary form of software requirement for a new software program underdeveloped. Use cases specify the expected behavior, but not the exact method of making it happen.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.



**DESCRIPTION**

**User**

The user has to register and create an account and also the user has to login with these credentials. Once, user login through an account he can ask for any kind of information about the vasavi college.

**System**

TheSystem responds according to the user's questions and gives the required information whatever he wants.

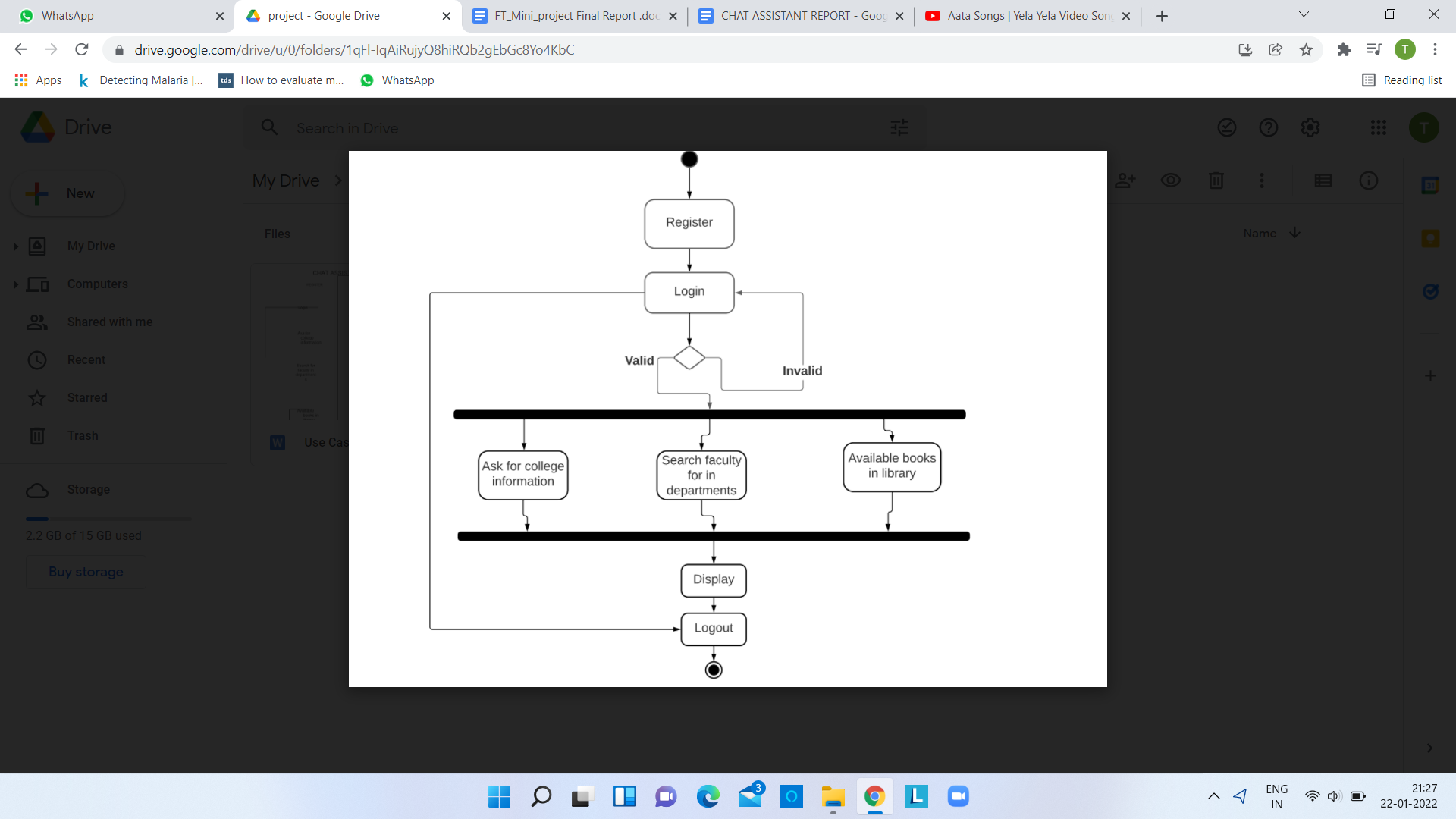
**Files**

The system stores all the registered accounts for the chat assistant permanently in the form of files.

**Logout**

It allows users to exit from the chat assistant.

**ACTIVITY DIAGRAM**

****

**IMPLEMENTATION**

Chat-assistant.py

from tkinter import \*

import os

#window for register

def register():

global reg\_screen

reg\_screen = Toplevel(fs)

reg\_screen.title("REGISTER")

reg\_screen.geometry("300x300")

global username

global password

global username\_entry

global password\_entry

username = StringVar()

password = StringVar()

Label(reg\_screen, text="PLEASE ENTER THE BELOW DETAILS TO REGISTER", bg="skyblue").pack()

Label(reg\_screen, text="").pack()

L1 = Label(reg\_screen, text="USERNAME")

L1.pack()

username\_entry = Entry(reg\_screen, textvariable=username)

username\_entry.pack()

L2 = Label(reg\_screen, text="PASSWORD")

L2.pack()

password\_entry = Entry(reg\_screen, textvariable=password, show='\*')

password\_entry.pack()

Label(reg\_screen, text="").pack()

Button(reg\_screen, text="SUBMIT", width=15, height=1, bg="skyblue", command=register\_user).pack()

#window for login

def login():

global login\_screen

login\_screen = Toplevel(fs)

login\_screen.title("LOGIN")

login\_screen.geometry("300x300")

Label(login\_screen, text="ENTER THE LOGIN DETAILS",bg ='skyblue',font = 20).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

L1 = Label(login\_screen, text="USERNAME").pack()

username\_login\_entry = Entry(login\_screen, textvariable=username\_verify)

username\_login\_entry.pack()

Label(login\_screen, text="").pack()

L2 = 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,bg='skyblue',command=login\_verify).pack()

# Implementing event on register button

def register\_user():

username\_info = username.get()

password\_info = password.get()

file = open(username\_info, "w")

file.write(username\_info + "\n")

file.write(password\_info)

file.close()

username\_entry.delete(0, END)

password\_entry.delete(0, END)

Label(reg\_screen, text="Registration Success", fg="green", font=("calibri", 11)).pack()

# Implementing event on login button

def login\_verify():

username1 = username\_verify.get()

password1 = password\_verify.get()

username\_login\_entry.delete(0, END)

password\_login\_entry.delete(0, END)

list\_of\_files = os.listdir()

if username1 in list\_of\_files:

file1 = open(username1, "r")

verify = file1.read().splitlines()

if password1 in verify:

login\_sucess()

else:

password\_not\_recognised()

else:

user\_not\_found()

# Designing popup for login success

def login\_sucess():

import main

main.get\_response("HI")

# Designing popup for login invalid password

def password\_not\_recognised():

global password\_not\_recog\_screen

password\_not\_recog\_screen = Toplevel(login\_screen)

password\_not\_recog\_screen.title("Dialog")

password\_not\_recog\_screen.geometry("250x200")

Label(password\_not\_recog\_screen, text="INVALID PASSWORD ").pack()

Button(password\_not\_recog\_screen, text="EXIT", command=delete\_password\_not\_recognised).pack()

# Designing popup for user not found

def user\_not\_found():

global user\_not\_found\_screen

user\_not\_found\_screen = Toplevel(login\_screen)

user\_not\_found\_screen.title("Dialog")

user\_not\_found\_screen.geometry("300x300")

Label(user\_not\_found\_screen, text="USER NOT FOUND").pack()

Button(user\_not\_found\_screen, text="EXIT", command=delete\_user\_not\_found\_screen).pack()

# Deleting popups

def delete\_login\_success():

login\_success\_screen.destroy()

def delete\_password\_not\_recognised():

password\_not\_recog\_screen.destroy()

def delete\_user\_not\_found\_screen():

user\_not\_found\_screen.destroy()

def first\_screen():

global fs

fs = Tk()

fs.geometry("350x280")

fs.title("CHATBOT LOGIN")

Label(text="SELECT YOUR CHOICE", bg="skyblue", width="250", height="2", font=("Calibri", 13)).pack()

Label(text="").pack()

Button(text="LOGIN", height="2", width="40", command=login).pack()

Label(text="").pack()

Button(text="REGISTER", height="2", width="40", command=register).pack()

fs.mainloop()

first\_screen()

Main.py

import re

import long\_responses as long

def msg\_probability(user\_msg, recognised\_words, single\_response=False, required\_words=[]):

msg\_certainity = 0

has\_required\_words = True

for word in user\_msg:

if word in recognised\_words:

msg\_certainity += 1

percentage = float(msg\_certainity) / float(len(recognised\_words))

for word in required\_words:

if word not in user\_msg:

has\_required\_words = False

break

if has\_required\_words or single\_response:

return int(percentage \* 100)

else:

return 0

def check\_all\_msg(msg):

highest\_list = {}

def response(assistant\_response, list\_of\_words, single\_response=False, required\_words=[]):

highest\_list[assistant\_response] = msg\_probability(msg, list\_of\_words, single\_response, required\_words)

#sample responses

response('Hello!', ['hello', 'hi', 'hey', 'sup', 'heyo'], single\_response=True)

response('See you!, Have a nice day!', ['bye', 'goodbye', 'logout'], single\_response=True)

response('You\'re welcome!', ['thank', 'thanks'], single\_response=True)

response('Vasavi College Of Engineering is located in Ibrahimbagh ,Hyderabad',['where', 'is','vasavi', 'vce', 'college','located'],required\_words=['located'])

response('A++ GRADING', ['what', 'is', 'the', 'naac', 'grading', 'of', 'vasavi'], required\_words=['grading', 'naac'])

response('201', ['what', 'is', 'nirf', 'ranking'], required\_words=['ranking', 'nirf'])

response('486', ['how', 'many', 'net', 'selections', 'in', 'placements'], required\_words=['how', 'selections', 'placements'])

response('82.65', ['what', 'is', 'the', 'percentage', 'of', 'selected', 'in', 'placement'], required\_words=['percentege', 'selected'])

response('Osmania University', ['vasavi', 'is', 'affiliated', 'to', 'vce'], required\_words=['affiliated'])

response('1981', ['vasavi', 'founded', 'in', 'which', 'year'], required\_words=['founded', 'year'])

response('private institute', ['which', 'type', 'of', 'institute'], required\_words=['type', 'institute'])

response('Dr.S.V.Ramana is the principal of vasavi college of engineering', ['who', 'is', 'the', 'principal', 'of', 'vasavi','vce'], required\_words=['principal'])

response('Dr.K.Ram Mohan rao is the HOD of IT Department', ['who', 'is', 'the', 'HOD', 'of', 'IT', 'department'], required\_words=['hod', 'it'])

response('Dr.B.Sridhar is the HOD of civil Department', ['who', 'is', 'the', 'HOD', 'of', 'civil', 'department'], required\_words=['hod', 'civil'])

response('Dr.E.Sreenivasa is the HOD of ECE Department', ['who', 'is', 'the', 'HOD', 'of', 'ECE', 'department'], required\_words=['hod', 'ece'])

response('Dr.M.Chakravarthy is the HOD of EEE Department', ['who', 'is', 'the', 'HOD', 'of', 'EEE', 'department'], required\_words=['hod', 'eee'])

response('Dr.T.Ram Mohan rao is the HOD of Mechanical Department', ['who', 'is', 'the', 'HOD', 'of', 'mech', 'department'], required\_words=['hod', 'mech'])

response('There are 8 departments in vce', ['how', 'many', 'departments', 'are', 'there', 'in', 'vasavi'],required\_words=['how', 'departments'])

response('OOP and DS', ['what', 'are', 'different', 'core','subjects', 'in', 'IT and CSE'], required\_words=['core','subjects', 'IT and CSE'])

response('Networks analysis and Electronic devices', ['what','are', 'different', 'core', 'subjects', 'in', 'ECE'],required\_words=['core', 'subjects', 'ECE'])

response('Electrical Network Analysis and Electromagnetic FielTheory', ['what','are', 'different', 'core', 'subjects', ' in ','EEE'], required\_words = ['core', 'subjects', 'EEE'])

response('Geology and Surveying', ['what', 'are', 'different','core', 'subjects', 'in', 'Civil'], required\_words=['core','subjects', 'Civil'])

response('Mechanics of Materials and Materials Engineering',['what', 'are', 'different', 'core', 'subjects', 'in','Mechanics'], required\_words = ['core', 'subjects', 'Mechanics'])

response('+914023146003', ['what', 'is', 'the','college', 'landline number', '?'], required\_words=['landline'])

response('Vasavi college gives admissions through EAMCET and JEE mains', ['what ', 'are', 'different', 'categories', 'of', 'admissions', 'available', 'in', 'vasavi' 'vce', 'college'], required\_words = ['admissions'])

response('https://vce.ac.in',['what','is','the','college','website'], required\_words = ['website'])

response('https://maps.app.goo.gl/GuDFXL4HpyQkic8m6',['what','is','the','route','map','for','vasavi','college', 'vce'], required\_words = ['map'])

response('VCE conducts online exams on moodle platform',['where','do','you','counduct','online','exams'], required\_words = ['online','exams'])

response('https://moodle.vce.ac.in',['what','is','the','link','for', 'moodle','platform'],required\_words = ['link', 'moodle'])

response('Mr. M. Ravi Kumar is the librarian of vce',['who', 'is', 'the', 'librarian', 'of', 'vasavi','vce'], required\_words=['librarian'])

response('+914023146095',['what', 'is', 'the','college', 'library', 'number'], required\_words=['library','number'])

response('library@staff.vce.ac.in',['what', 'is', 'the', 'college', 'library', 'email'], required\_words = ['library', 'email'])

response('500089 is the pincode of vce',['what','is','the','pincode','of','vasavi','college'], required\_words = ['pincode'])

response('https://www.vce.ac.in/Facilities/Library/Working\_Hours', ['what', 'are', 'the', 'library', 'timings'], required\_words=['library','timings'])

response('https://www.vce.ac.in/Facilities/Library/Rare\_Books',['what', 'are', 'the', 'rare','books','in', 'library'], required\_words = ['rare','books'])

response('https://www.vce.ac.in/Facilities/Library/Library\_Rules',['what', 'are', 'the', 'rules', 'of', 'vce', 'library'], required\_words=['rules', 'library'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=Civil',['civil','faculty'], required\_words=['civil','faculty'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=CSE', ['CSE', 'faculty'], required\_words=['cse', 'faculty'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=ECE', ['ECE', 'faculty'], required\_words=['ece', 'faculty'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=EEE', ['EEE', 'faculty'], required\_words=['eee', 'faculty'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=Mech.', ['mechanical', 'faculty'], required\_words=['mech', 'faculty'])

response('https://www.vce.ac.in/Staff\_List.cshtml?Department=IT', ['IT', 'faculty'], required\_words=['it', 'faculty'])

response('https://www.vce.ac.in/Facilities/GamesnSports/Facilities\_Games',['facility','of','games'], required\_words=['games'])

response('Sri Pendekanti Venkatasubbaiah is the founder of vce',['vce','founder', 'vasavi'], required\_words=['founder'])

response('No in vasavi college we don\'t have hostel facility', ['Is', 'there', 'any', 'hostel', 'facility', 'available', 'in', 'vasavi', 'college', 'vce'], required\_words=['hostel', 'facility'])

response('Yes we have transportation facility in vasavi college', ['what', 'are', 'transportation', 'facilities', 'in', 'vasavi', 'college', 'vce'], required\_words=['transportation'])

response('https://www.vce.ac.in/Facilities/Transport.cshtml', ['what', 'are', 'different', 'routes', 'of', 'transportation'], required\_words=['routes'])

best = max(highest\_list, key=highest\_list.get)

return long.unknown() if highest\_list[best] < 0.5 else best

def get\_response(user\_input):

split\_msg = re.split(r'\s+|[,;?!.-]\s\*', user\_input.lower())

response = check\_all\_msg(split\_msg)

return response

print("\*\*\*\*\*\*\*\* WELCOME TO THE VASAVI CHAT ASSISTANT \*\*\*\*\*\*\*\*")

while True:

s = get\_response(input('You: '))

print('Bot: ' + s)

if s == 'See you!, Have a nice day!':

break

Long\_responses.py

import random

def unknown():

response = ["Could you please re-phrase that? ",

"Sorry i didn't get you."][

random.randrange(2)]

return response

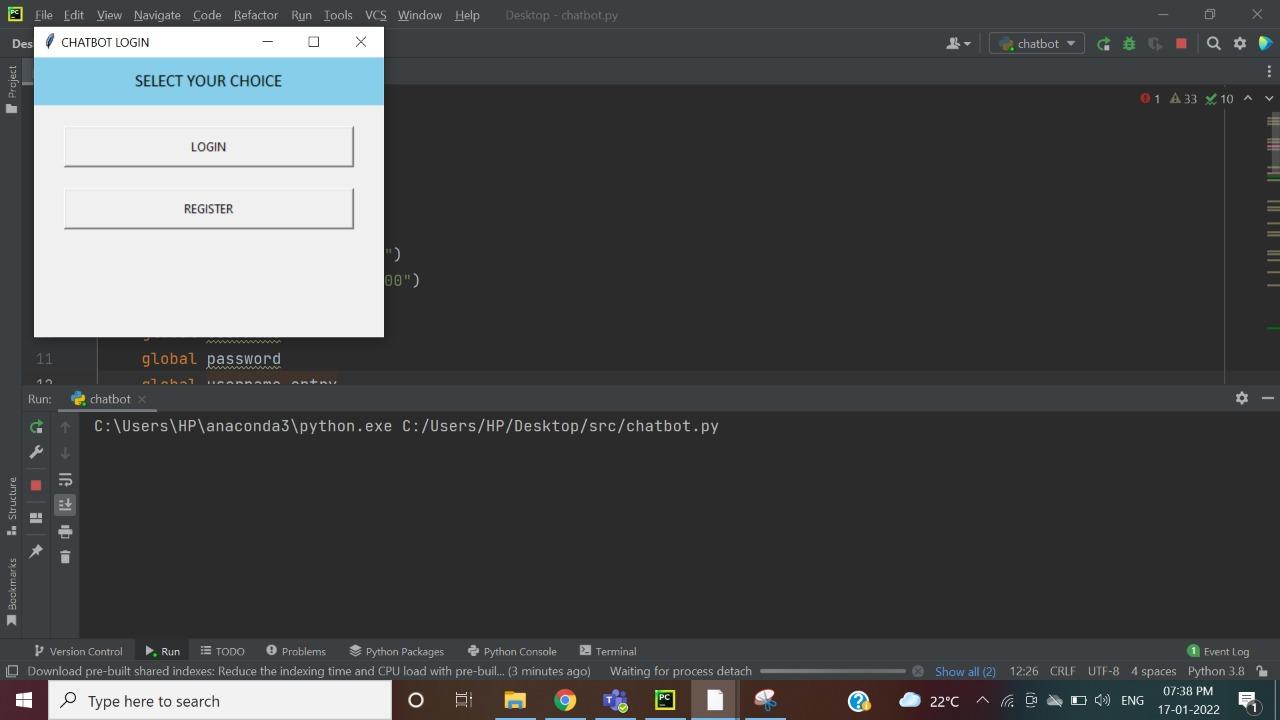
**c.Testing**

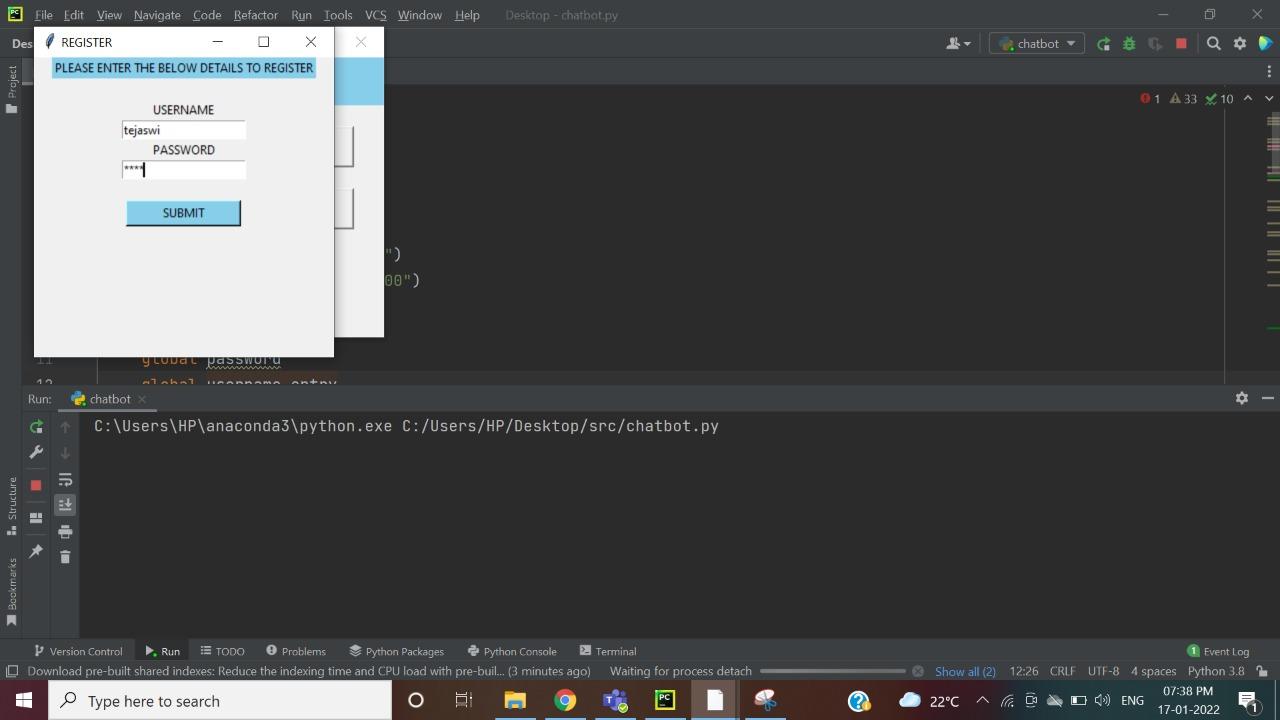
|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
| **Test Case Id:** TC01 | | **Use Case ID:** UC01 | |
| **Test Case Title:** Register | |  | |
| **Test Case Description:**  User have to register to the chat assistant with the username and password | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.Open the chat assistant using any editor and run.  2.Select Register from the ui  3.Give appropriate username and password in the pop-up ui. | The user gets an account which requires username and password to access it. | | The user can successfully access the account with the given credentials. |

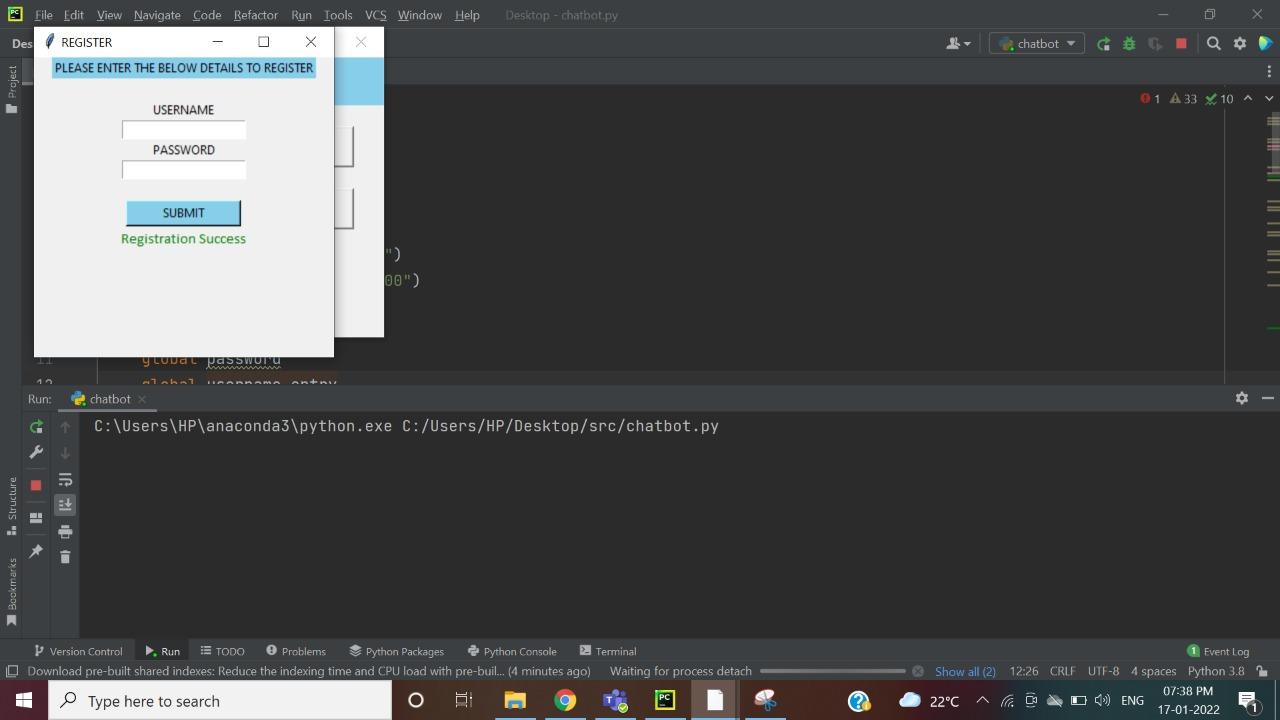
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | | | |  |
| **Test Case Id:** TC02 | | **Use Case ID:** UC02 | |  |
| **Test Case Title:** Login | |  | |  |
| **Test Case Description:**  Now users can login with the credentials. | |  |
| **Test Steps** | **Expected Result** | | **Actual Result** |  |
| 1.Open the chat assistant using any editor and run.  2.Select Login from the ui  3. Fill the username and password with the credentials. | If the user gives a valid username and password then they can login to the chat assistant. | | If the user gives a valid username and password then they can login to the chat assistant. Otherwise, a dialog box will pop-up which asks to give valid data. | |

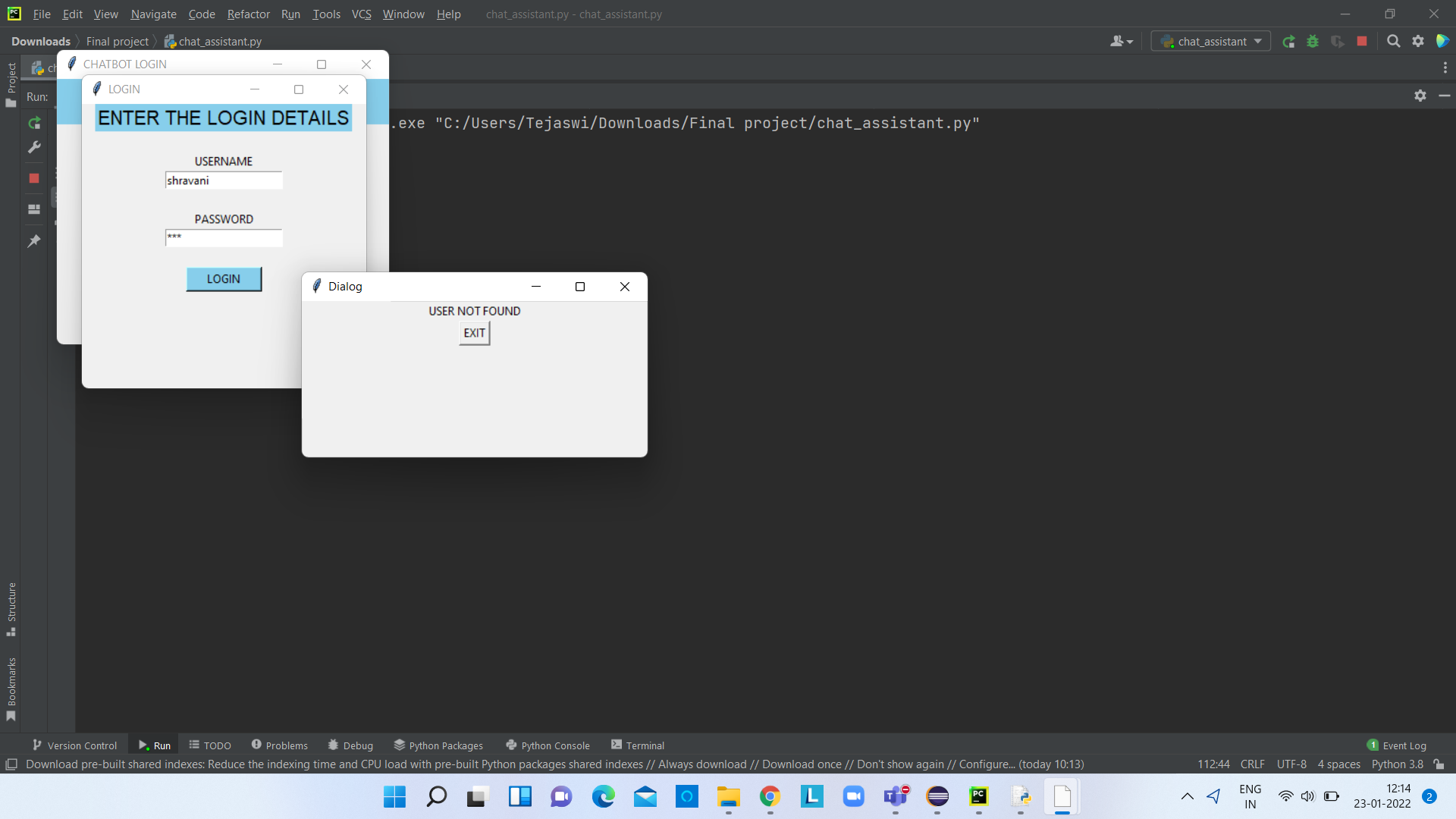
|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | | | |
| **Test Case Id:** TC03 | | **Use Case ID:** UC03 | |
| **Test Case Title:** Chat Assistant | |  | |
| **Test Case Description:**  After giving correct credentials, one can login to the chat assistant. | |
| **Test Steps** | **Expected Result** | | **Actual Result** |
| 1.Open the chat assistant using any editor and run.  2.Select Login from the ui and login with the credentials.  3. After that, one can access the chat assistant. | The user gets access to the chat assistant and can ask any number of questions about vasavi college and the system gives appropriate responses. | | The user successfully accessed the chat assistant and can ask the questions related to the vasavi college and it gives appropriate responses. |

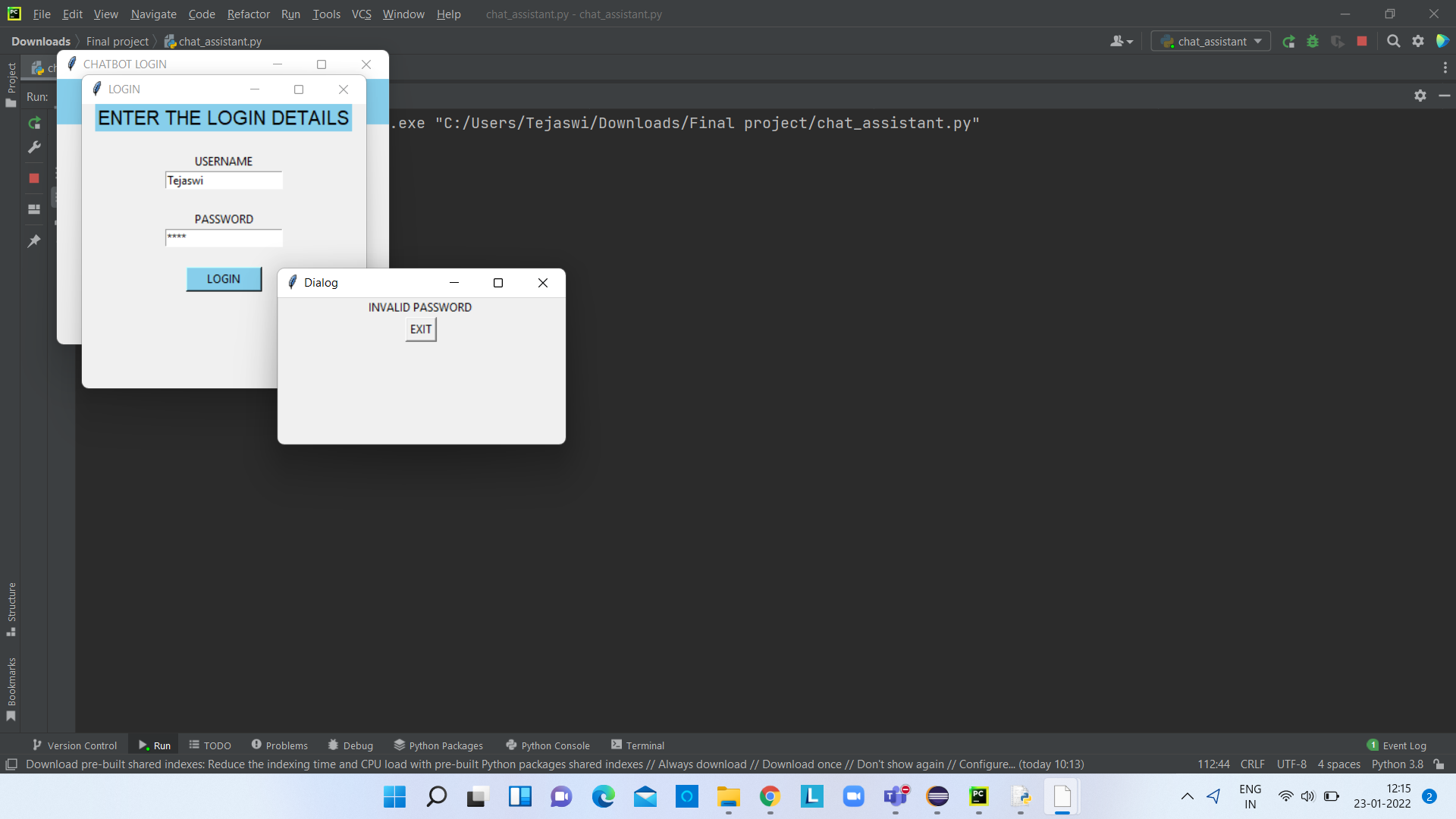
**RESULTS**

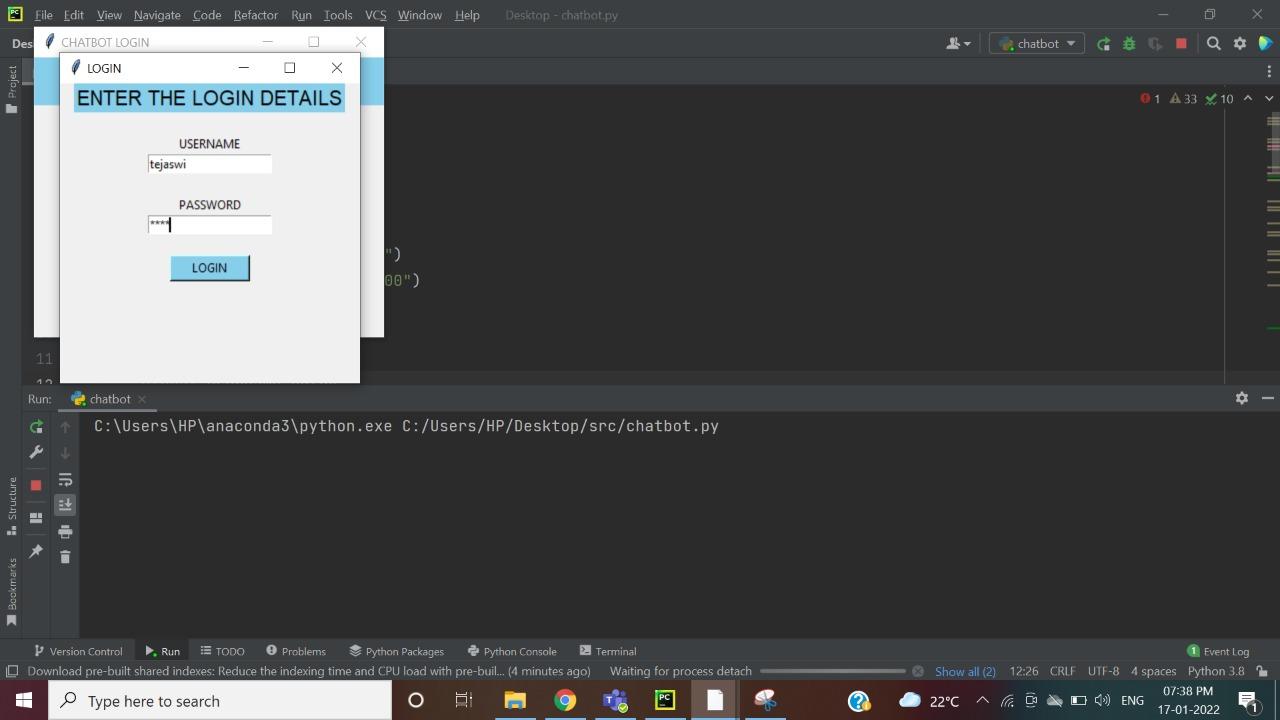
****

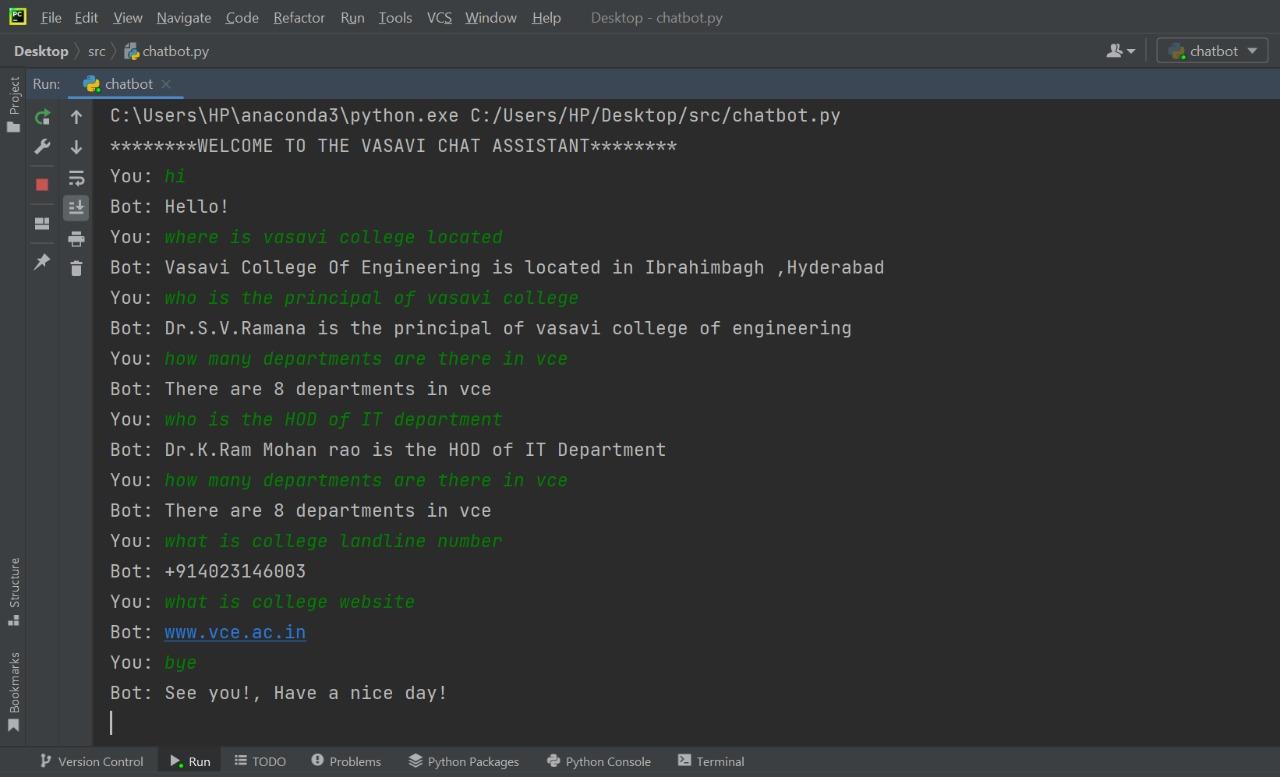
****

****

****

****

****

****

**Outcomes from the project apart from curriculum**

● I have got to know how to use different kinds of modules.

●I have discovered different kinds of libraries and modules and got to know their implementation and working.

●I have got to know about re modules and how to use the regular expressions.

●I have learnt how to build a GUI using the tkinter module.

●I have got to know how to handle events in GUI and about the components in them.

**CONCLUSION AND FUTURE WORK:**

As we know college websites are more beneficial to enquire or to find any information like announcements. It's quite useful nowadays. In general if we want to know any particular information about colleges then we can directly use these kinds of websites or chat assistants.

So, this kind of chat assistant has a lot of scope in the future world.

**Future work**

● These chat assistants become faster and more accurate.

● Improve GUI for the chat assistant.

● Taking user responses through both voice and chat gives appropriate responses as the combination of both chat and voice bots.

The project is successfully implemented with all the features and also accessing this chat assistant is very simple.

**11. REFERENCES**

1. <https://youtu.be/Xt6SqWuMSA8>
2. https://online.datasciencedojo.com/blogs/building-a-rule-based-chatbot-in-python