COMPANION INTELLIGENT CHATBOT

A Main Project Report

submitted by

GAYATHRI K(MES20MCA-2019)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

July 2022

DECLARATION

I undersigned hereby declare that the project report **COMPANION INTELLIGENT CHATBOT**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Prof.Hyderali K, Associate Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: KUTTIPPURAM

Date:

GAYATHRI K(MES20MCA-2019)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **COMPANION INTELLIGENT CHATBOT** is a bona fide record of the Main Project work carried out by **GAYATHRI K(MES20MCA-2019)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

My endeavor stands incomplete without dedicating my gratitude to a few people who have contributed towards the successful completion of my project.I pay my gratitude to the Almighty for his invisible help and blessing for the fulfillment of this work.At the outset I express my heartful thanks to my project coordinator Ms.Priya J D for his valuable guidance and supervision. I take this opportunity to express my profound gratitude to Prof.Hyderali K, my guide for his valuable support, timely advise and strict schedules to complete my project. I also grateful to all my teaching and non-teaching staff for their encouragement, guidance and whole-hearted support. Last but not least, I gratefully indebted to my family and friends, who gave us a precious help in doing my project.

GAYATHRI K(MES20MCA-2019)

Abstract

Today online Social Network Mental Disorder Detection (SNMDD) are usually treated at a late stage. To address this issue, we propose an approach, new to the current practice of SNMD detection, by mining data logs of OSN users to actively identify potential SNMD cases early. We develop a machine learning framework for detecting SNMDs, namely Social Network Mental Disorder Detection (SNMDD). Moreover, we design and analyze many features from OSNs, such as parasociality, self-disclosure, etc., which serve as important factors or proxies for identifying SNMDs. The proposed framework can be deployed as a software program to provide an early alert for potential patients and their advisors. The System also provides an emotion graph which helps to know about the emotion levels of various users.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Open Sourcing Mental illness (OSMI) is a non-profit organization that assist awareness about mental health in workplaces and also suggest workplaces to identify the best resources so that they can provide best atmosphere to their employees. In 2017, a survey conducted by OSMI mental health in workplace where 750 responses taken from various employees who were working in range of tech divisions. In this individual's personal and professional factors were taken. And used this survey's dataset to trained different ML models to analyse and find the patterns of stress and mental disorders .The models used in machine learning to detect stress and anxiety based on datasets are: Logistic regression, KNN classifier, Decision Tree, random forest classifier, boosting and bagging. It is found that people working in tech company were in slightly high danger of developing stress as compared to others. To make or provide good atmosphere in workplace Better HR policies should be provided to employees.

1.1.1 Motivation

My work is to make a chatbot that will talk to user. It will ask questions from the user and give him choices for the question or he can write them. According to the chosen answers it will predict the type of stress he/ she is facing and respective of that it will show management of the stress first I researched and looked for the chatbot and how I will implement prediction. I read research papers and looked on the internet for the following.

Decision tree is a very powerful and popular tool for classification and prediction. It may be a flowchart like tree type structure, where every single node represents a test on an attribute, each branch represents a result/outcome of the test, and every leaf node (terminal node) holds a category label. It works as by selecting the attributes which are the nodes. If the attributes and selected then those nodes are implemented and hence goes on in the tree. So, this is the decision tree which I have created for our work and implemented for the Depression only.

1.2 Objective

it will give user the ability to identify the stress based on the given questions and answers.I have chosen this project because I want to provide user more easy, user-friendly, interactive way to find out about their stress.

This approach is for the commercial purpose where I ditch the old system of given set of questions and show results. Here it is more user friendly and can be integrated in any website easily. User talk to the bot and first bot get his details and asks him to give the test for the prediction. When the stress is predicted, according to that it gives management for that. I have applied machine learning techniques for the prediction of stress.I have made our own decision tree for the prediction of stress and according to that decision node will give the management of the stress For chatbot I have used Rasa framework, as it was easy to use and understand.

1.3 Contribution

Stress becomes a major issue in today's time and also lead to many health problems of People of different professions, lifestyles, gender and age groups. My work is to make a chatbot that will talk to user. It will ask questions from the user and give him choices for the question or he can write them. According to the chosen answers it will predict the type of stress he/ she is facing and respective of that it will show management of the stress.

1.4 Report Organization

The project report is divided into five chapters.Chapter 1 describes introduction.Chapter 2 describes literature survey. Chapter 3 describes the methodology used for implementing the project. Chapter 4 gives the results and discussions. Finally Chapter 5 gives the conclusion.

Chapter 2

Literature Survey

From several studies it is found that employees working in IT professionals in industry are facing issues related to stress disorders. This is because of change in their work culture and their changing lifestyles.Many businesses and companies provide their workers with mental health programs to relieve their tension and the working environment. But these issues are too huge to handle and far from control. Machine Learning techniques are used to study the working in adults and to slim down the issues that powerfully determine the stress levels.According to OSMI mental health survey, It is a non-profit organization that promotes awareness about stress, mental illness and disorders in workplace and also helps in identifying best resources to help their employees. OSMI mental health organization, takes data from datasets from surveys and examine the patterns of stress and mental health disorders and to regulate the significant issues that contribute to same. Many ML techniques are used for these surveys such as: Logistic regression, KNN Classifier, Decision Tree, Boosting, Bagging and Random forest classifier. To increase the efficiency of these stress predicting models, we can use 'Naïve Bayes classifier'.

Stress is considered as a major problem for today's generation. It mostly arises in the field of Jobs because of the negative atmosphere. Employees working in different professions feel stress because of long working hours, work overload, time pressure, lack of variety, lack of breaks and almost all people of all age groups feel overstressed and suffering from many health problems. Stress also contributes in direct illness and through maladaptive health behaviors such as poor eating habits and smoking etc. So, management of stress should be started before the stress starts causing illness. According to (APA survey 2004), it is found that one in four

employees has taken a leave from their work to cope with their stress. Although many sensor technologies are developing to measure physical symptoms which reflects the stress level. Stress is predicted using data mining techniques by collecting the data of employees while working hours and during their meetings and while doing their pre planned work etc.

Stress is a significant occupational health concern at the workplace. In the recent studies there are various sensing modalities which help to model stress behavior based on non-obtrusive data.[10] An approach based on combination of multiple techniques like ensemble methods, semi-supervised learnings and transfer learning to build a model of subject with scarce data. A real-life, unconstrained study administered with 30 employees within two organizations. The results show that using information (instances or model) from similar subjects can improve the accuracy of the themes with scarce data. Transfer learning from different subjects can have a negative effect on accuracy. This increased the accuracy by 1071.58technique.

In this prediction of mood can done using data collected by using wearable sensor devices, smartphones and some clinical applications. To overcome the problems of lack of accuracy and reliable performance required for real world applications for collection of data, multi task learning (MLT) and Domain Adaption (DA) techniques are used to train the ML models. Three formulations of MLT are compared :-

1. Multi task deep neural network
2. Multi task kernel learning
3. Hierarchical Bayesian model

Empirical results based on real world data shows new modified models such as MTL deep neural network and Gaussian process with DA outdo their generic counterparts and the odds of predicting continuous amounts of tomorrow's recorded stress and physical wellbeing are also improved, based on data from today.

Chapter 3

Methodology

3.1 Introduction

Language plays an import role in the field of communication, as through languages you can express your feeling emotions etc. Emotion involves in feelings, behaviors, experience and cognitions. An emotion could be any strong feelings through some circumstances or mood or relationship. Exchange of emotion can be done through text, feelings, speech, video, audio etc.Human can recognize their feelings, emotions but this is a challenge that how a system recognize humans feelings in the form of text, video, audio.Here I propose a system (Application) that recognize the emotion of the humans from their text. The system will monitor the messages shared by users. From these information users mental disorder can be find out .And result may forward to their relatives if any negative thought may detected.

Today online Social Network Mental Disorder Detection (SNMDD) are usually treated at a late stage. To address this issue, we propose an approach, new to the current practice of SNMD detection, by mining data logs of OSN users to actively identify potential SNMD cases early. We develop a machine learning framework for detecting SNMDs, namely Social Network Mental Disorder Detection (SNMDD). Moreover, we design and analyze many features from OSNs, such as parasociality, self-disclosure, etc., which serve as important factors or proxies for identifying SNMDs. The proposed framework can be deployed as a software program to provide an early alert for potential patients and their advisors. The System also provides an emotion graph which helps to know about the emotion levels of various users.

I have chosen this project because we want to provide user more easy, user-friendly, in-

teractive way to find out about their stress. Bot will ask them few questions and according to the chosen answers, bot will show them the result and management of stress . Chatbot is a program that is designed to stimulate conversations or talk with human users. In this User communicates with the bot either with text or voice. Chatbot give logical reply or the already stored answers to the user for the questions. It is just like a user is talking to the other person. They interpret the questions and provide answer to them. They are used in messaging apps, e- commerce, etc. They can be customized for either single company based or public based. Stress is a physiological kickback to the social, behavioral or other physical issues that people face in their real-life activities, including in their environments like workplace, household, etc. Continued stress consumption can lead to some serious and extreme health issues, such as causing physical illness through its physiological consequences, changes in behavior, and problems with social isolation. Stress affects tons of individuals in their life like mood, behavior, health and quality of life. Disorders include: Headaches, heart attacks, depression, and system abnormalities include illnesses. They further note that, relative to the opposite forms of stress, acute stress appears to attract less consideration from researchers. Acute stress is also caused by multiple kinds of immediate emotional and physical challenges that people can encounter in several ways during ongoing everyday circumstances. This triggers certain physical changes, such as an increase in the heart, and these changes can induce severe long- term diseases and have a negative effect on the emotional and physical well-being of a person.

3.2 Algorithm Used

POLARITY ANALYZER ALGORITHM

Process of computationally identifying and categorizing opinions from piece of text,and determine whether the writer's attitude towards a particular topic or the product,is positive,negative or neutral.

Step 1:Tokenization(Dividing a para in to a different set of statements.)

Example:The Movie Was Great!

Step 2:Cleaning the Data(Remove the special characters.)

Step 3:Removing Stop Words.

Step 4:Classification

Step 5:Apply Supervised algorithm for classification.

Step 6:Calculation

eg:-+1+0=1

Since the Polarity is greater than 0 So the given statement is positive.(The Movie Was Great.)

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY ALGORITHM(TF-IDF)

Sentence 1:He is a good boy.

Sentence 2:She is a good girl.

Sentence 3:Boy and girl are good.

After removing stop words,

Sent 1:good boy

Sent 2:good girl

Sent 3:Boy girl good

Frequency of word "Good"=3

Frequency of word "Boy"=2

Frequency of word "Girl"=2

TF = No: of repetition of words in sentence/No: of words in Sentence.

IDF = log[No: of Sentences/No: of Sentences containing words]

3.3 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an enduser perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall.The user story of system is given in below Table 3.1.

User Story Id	**As a type of User**	**I want to ⟨perform some task⟩**	**So that I can ⟨Achive Some Goal⟩**
1	Admin	login	login successful height2
Admin	Verify councellors	can approve registered cxouncellors	
3	Admin	View emotion graphs	can view person's emotion status
4	Admin	View feedback	can view user feedback
5	User	Registration	user's can register with this app
6	user	Add post	users can add post
7	User	Send and accept friend request	User can send and accept friend request
8	User	Get counselling tips	user can get suggestions from counsello
9	User	Add feedback	user can add feedback
10	Counsellor	Registration	councillors can register with this app
11	Counsellor	view emotion graphs	can view person's emotion status
12	Counsellor	Provide counselling tips	counsellor can give suggestions for user

Table 3.1: User Story

3.4 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given in below Table 3.2.

User Story Id	Priority ⟨High-/Medium/Low ⟩	Size(Hours)	Sprint ⟨⟩	Status⟨Planned/In progress/Completed⟩	Release Date	Release Goal
1	Medium	2	1	completed	20/04/2022	Table desgn
2	High	3	1	completed	22/04/2022	Form design
3	High	5	1	completed	23/04/2022	Basic coding
4	High	5	2	completed	23/04/2022	creation of Datasets
5	Medium	5	2	completed	23/04/2022	Simulate Conversation with human users
6	High	5	3	completed	29/05/2022	Counsellor
7	High	5	3	completed	30/05/2022	Machine learning
8	Medium	5	4	completed	05/06/2022	Testing data
9	High	5	4	completed	06/07/2022	Output generation

Table 3.2: Product Backlog

3.5 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project paln is given in the below Table 3.3. The Project has four sprints:

1. Sprint 1 : Registration of Users and Authenticating the users

Two tasks are planned in this sprint. One is to register the users into the system and another one is to authenticate the registered users. The designing of UIS, coding, testing and validation of these functionality is be completed.

2. Sprint 2 : Login to account and Reset password

Plan for two functionalities is login and reset password are made in this sprint. One is to the login of the users into the system. Another one is the password recovery option, which

means the user can recover his/her password in case of any trouble. The UI designing, table designing, coding, testing also planned to complete.

3. Sprint 3 : view feedback,view user details,and view complaint and send reply

This sprint plan has three core functionalities. The first one is the major part of the system, which is the view user feedback.The second one is the view user details, And the third one is view complaint and send reply to user. The designing of UIS, table creation, coding, testing and validation for the three functionalities is also planned.

4. Sprint 4 : Add complaint and view reply,Add feedback,and chatbot

Sprint 4 is planned to complete three tasks. The first one is to Add complaint and view reply,Second one is add feedback and another one is chatbot. The UI designing, coding, testing and validation also planned to complete.

User Story Id	**Task Name**	**Start Date**	**End Date**	**Hours**	**Status**
1	Sprint 1	20/04/2022	20/04/2022	8	Completed
2	Sprint 1	4/05/2022	15/05/2022	10	Completed
3	Sprint 2	17/05/2022	28/05/2022	6	Completed
4	Sprint 2	29/05/2022	1/06/2022	5	Completed
5	Sprint 3	2/06/2022	5/06/2022	5	Completed
6	Sprint 4	6/06/2022	20/06/2022	11	Completed
7	Sprint 5	20/06/2022	24/06/2022	4	Completed
8	Sprint 5	25/06/2022	29/06/2022	4	Completed

Table 3.3: Project Plan

3.6 Sprint Backlog

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.Sprint Backlog is given in the below Table 3.4.

Backlog Item	**Status And Completion Date**	**Original Estimation in Hours**	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**	**Day 11**	**Day 12**	**Day 13**	**Day 14**
UserStory1,2,3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table 0	Form Design	1/05/2022	8	1	1	1	2	0	1	1	0	0	1	0	0	0
Coding	15/05/2022	10	1	3	0	1	1	0	0	3	0	1	0	0	0	0
UserStory 4,5																
Creation of datasets	28/05/2022	6	1	1	1	1	2	0	0	0	0	0	0	0	0	0
Simulate conversation with human users	1/06/2022	5	0	4	1	0	0	0	0	0	0	0	0	0	0	0
UserStory 6,7																
Counsellor	5/06/2022	3	0	0	0	0	1	1	1	0	0	0	0	0	0	0
Machine learning	20/06/2022	11	1	2	1	2	1	1	1	0	0	0	2	0	2	1
UserStory 8,9																
Testing data	24/06/2022	4	0	0	0	0	1	2	1	0	0	0	0	0	0	0
Output generation	29/06/2022	4	0	0	2	0	0	0	0	0	0	2	0	0	0	0
Total		51	4	11	5	6	4	6	4	0	1	5	2	0	2	1

Table 3.4: Sprint BackLog

3.7 Sprint Backlog (Actual)

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

Backlog Item	**Status And Completion Date**	**Original Estimation in Hours**	**Day 1**	**Day 2**	**Day 3**	**Day 4**	**Day 5**	**Day 6**	**Day 7**	**Day 8**	**Day 9**	**Day 10**	**Day 11**	**Day 12**	**Day 13**	**Day 14**
UserStory1,2,3			hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Form 0	table Designing	1/05/2022	9	1	1	1	2	0	1	1	0	2	0	0	0	0
Coding	15/05/2022	10	1	3	0	1	1	0	1	0	0	3	0	0	0	0
UserStory 4,5																
Creation of datasets	28/05/2022	6	1	1	0	1	1	2	0	0	0	0	0	0	0	0
Simulate conversation with human users	1/06/2022	6	0	4	2	0	0	0	0	0	0	0	0	0	0	0
UserStory 6,7																
Counsillor	5/06/2022	4	0	0	0	0	2	1	1	0	0	0	0	0	0	0
Machine learning	20/06/2022	13	1	2	1	2	0	2	0	0	0	0	2	0	2	1
UserStory 8,9																
Testing data	24/06/2022	4	0	0	0	0	1	2	1	0	0	0	0	0	0	0
Output generation	29/06/2022	4	0	0	2	0	0	0	0	0	0	2	0	0	0	0
Total		53	4	11	5	6	4	8	4	0	1	5	2	0	2	1

Table 3.5: Sprint Actual

3.8 Database and User Interfaces

The data that are captured through this system using different user friendly graphical user interfaces which are shown in Appendix (Fig A.1 to A.6). The data captured are stored in MySQL database and the database description are shown in Appendix (), which gives the details of the type of data and other characteristics of the data. Sample source code is also attached in Appendix.

3.9 Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. To show the continuous development of the project the Gitlab histories is shown in Appendix.

Chapter 4

Results and Discussions

4.1 Datasets

The term data set refers to a file that contains one or more records. The record is the basic unit of information used by a program running on OS.

Any named group of records is called a data set. Data sets can hold information such as medical records or insurance records, to be used by a program running on the system. Data sets are also used to store information needed by applications or the operating system itself, such as source programs, macro libraries, or system variables or parameters. For data sets that contain readable text, you can print them or display them on a console (many data sets contain load modules or other binary data that is not really printable). Data sets can be cataloged, which permits the data set to be referred to by name without specifying where it is stored.

In simplest terms, a record is a fixed number of bytes containing data. Often, a record collects related information that is treated as a unit, such as one item in a database or personnel data about one member of a department. The term field refers to a specific portion of a record used for a particular category of data, such as an employee's name or department. The records in a data set can be organized in various ways, depending on how we plan to access the information. If you write an application program that processes things like personnel data, for example, your program can define a record format for each person's data.

There are many different types of data sets in OS, and different methods for accessing them. Among the most commonly used types are:

Sequential

In a sequential data set, records are data items that are stored consecutively. To retrieve the tenth item in the data set, for example, the system must first pass the preceding nine items. Data items that must all be used in sequence, like the alphabetical list of names in a classroom roster, are best stored in a sequential data set.

Partitioned

A partitioned data set or PDS consists of a directory and members. The directory holds the address of each member and thus makes it possible for programs or the operating system to access each member directly. Each member, however, consists of sequentially stored records. Partitioned data sets are often called libraries. Programs are stored as members of partitioned data sets. Generally, the operating system loads the members of a PDS into storage sequentially, but it can access members directly when selecting a program for execution.

VSAM

In a Virtual Storage Access Method (VSAM) key sequenced data set (KSDS), records are data items that are stored with control information (keys) so that the system can retrieve an item without searching all preceding items in the data set. VSAM KSDS data sets are ideal for data items that are used frequently and in an unpredictable order.

4.2 Results

This project gives us the information about the psychological diseases and the effect it is giving to us and through a chatbot we can predict those and give management to it. Stress, anxiety, depression, phobia and insomnia were mentioned in this project. Decision tree was used to design the model and each leaf node provided us with the management information.We can provide suggestions to the patients based on their emotions.

Chapter 5

Conclusions

This Paper gives us the information about the psychological diseases and the effect it is giving to us and through a chatbot we can predict those and give management to it. Stress, anxiety, depression, phobia and insomnia were mentioned in this paper. Decision tree was used to design the model and each leaf node provided us with the management information. We also gave number to measure the range of the stress i.e. 0,1 is for Low Depression 2,3 for medium Depression 4 is for high depression We were able to make a working prototype of our approach. In this we only mentioned about Depression and if the person is having high, medium or low. We have given pre-defined buttons for the ease of User. They just need to click on the button and it will store that and move ahead.

References

[1] **Andrew, Ng** (2011) Sparse Autoencoder, [Online]. Available: `https://web.stanford.edu/class/cs294a/sparseAutoencoder_2011new.pdf`.

[2] **Krogh, A. and Hertz, J.A.** (1992) A simple weight decay can improve generalization, Advances in Neural Information Processing Systems, 950-957.

[3] **Gonzalez, R.C. and Wintz, P.** Digital Image Processing, 2nd Edition, Addison-Wesley, 1987.

[4] **Bazen, A.M. and Gerez, S.H.** (2001) Segmentation of fingerprint images, *ProRISC 2001 Workshop on Circuts, Systems and Signal Processing*, Veldhoven, The Netherlands.

Appendix

Source Code

webcode.py

```
  from flask import *
from src.dbconnect import *
app=Flask(__name__)
app.secret_key="aaaaaq"
import functools
def login_required(func):
   @functools.wraps(func)
   def secure_function():
      if "lid" not in session:
         return redirect ("/")
      return func()
   return secure_function
@app.route('/')
def login():
   return render_template("login.html")


@app.route('/viewuser')
@login_required
def viewuser():
   qry = "SELECT * FROM users"
   s = select(qry)
   return render_template("viewuser.html",val=s)



@app.route('/councviewuser')
@login_required
def councviewuser():
   qry = "SELECT * FROM users"
   s = select(qry)
   return render_template("councviewuser.html",val=s)




@app.route('/councviewuser1')
@login_required
def councviewuser1():
   id=request.args.get("id")
   session['uuid']=id
```

Appendix

```
    qry="SELECT SUM(`emotion`),COUNT(`emotion`)-SUM(`emotion`) FROM `postemo` JOIN `post` ON `post`.`Pid`=`postemo`.`pid`
          WHERE `post`.`Userid`=%s"
    res=selectonecond(qry,id)
    session['uid']=id
    rr=""
    print(res,"=========")
    if str(res[0])=="None":
        rr= "na"
    else:
        pc=[int(res[0]),int(res[1])]
        x=['positive','negative']

        import matplotlib.pyplot as plt
        import datetime



        # giving the values against
        # each value at x axis
        y = pc
        plt.bar(x, y)

        # setting x-label as pen sold
        plt.xlabel("Emotion")

        # setting y_label as price
        plt.ylabel("Count")
        plt.title(" Emotion Graph")
        fn=datetime.datetime.now().strftime("%Y%m%d%H%M%S")+".png"
        plt.savefig("static/graph/"+fn)
        rr=fn
        plt.close()
    print(rr,"++++++++++++++++")
    return render_template("councviewuser1.html",val=rr)
```

```
@app.route('/reply')
@login_required
def reply():
    id=request.args.get('id')
    session['cid']=id
    return render_template("reply.html")

@app.route('/inssugg',methods=['post'])
def inssugg():
    sug=request.form['s']
    qry="INSERT INTO `suggestion` VALUES(NULL,%s,%s,%s,CURDATE())"
    val=(session['uuid'],session['lid'],sug)
    iud(qry,val)
    return '''<script>alert('added');window.location='/councviewuser'</script>'''


@app.route('/reply1',methods=['post'])
def reply1():
    reply=request.form['textarea']
    qry="update complaint set reply=%s where cid=%s"
    val=(reply,session['cid'])
```

```
    iud(qry,val)
    return '''<script>alert('added');window.location='/viewcomplaint'</script>'''


@app.route('/viewcomplaint')
@login_required
def viewcomplaint():
    qry="SELECT `users`.`fname`,`users`.`lname`,`complaint`.* FROM `complaint` JOIN `users` ON
          `complaint`.`user_lid`=`users`.`lid` where complaint.reply='pending'"
    res=select(qry)
    return render_template("viewcomplaint.html",val=res)


@app.route('/viewfeedback')
@login_required
def viewfeedback():
    qry="SELECT `users`.`fname`,`users`.`lname`,`feedback`.* FROM `feedback` JOIN `users` ON
          `users`.`lid`=`feedback`.`user_lid`"
    s = select(qry)
    return render_template("viewfeedback.html",val=s)

@app.route('/homepage')
@login_required
def homepage():
    return render_template("homepage.html")







@app.route('/login2',methods=['post'])
def login2():
    uname=request.form['textfield']
    pword=request.form['textfield2']
    q="select * from login where username=%s and password=%s"
    val=(uname,pword)
    s=selectonecond(q,val)
    if s is None:
        return '''<script>alert('Invalid user name or password');window.location='/'</script>'''
    elif s[3]=='admin':
        session['lid']=s[0]
        return '''<script>alert('login successfully');window.location='/homepage'</script>'''
    elif s[3] == 'counsellor':
        session['lid'] = s[0]
        return '''<script>alert('login successfully');window.location='/councilhome'</script>'''
    else:
        return '''<script>alert('Invalid user name or password');window.location='/'</script>'''



@app.route("/logout")
def logout():
    session.clear()
    return render_template("login.html")

@app.route('/councilhome')
def councilhome():
    return render_template("counscellorhomepage.html")

@app.route('/addcounsellor')
def addcounsellor():
    return render_template("Counsellor_registration.html")
```

Appendix

```
@app.route('/councellor_reg',methods=['post'])
def councellor_reg():
    fname=request.form['textfield']
    lname = request.form['textfield2']
    dob=request.form['textfield3']
    gender=request.form['radiobutton']
    place = request.form['textfield4']
    post = request.form['textfield5']
    pincode = request.form['textfield6']
    qualification = request.form['textfield7']
    phoneno = request.form['textfield8']
    emailid = request.form['textfield9']
    username = request.form['textfield10']
    password = request.form['textfield11']
    qry="INSERT INTO `login` VALUES(NULL,%s,%s,'pending')"
    val=(username,password)
    loginid=iud(qry,val)
    qry1="INSERT INTO `counsellor` VALUES(NULL,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
    val2=(loginid,fname,lname,dob,gender,place,post,pincode,qualification,phoneno,emailid)
    iud(qry1,val2)
    return '''<script>alert('Successfully Registered');window.location='/'</script>'''

@app.route('/approve_counc',methods=['post','get'])
def approve_counc():
    qry="SELECT `counsellor`.* FROM `counsellor` JOIN `login` ON `login`.`id`=`counsellor`.`login_id` WHERE
        `login`.`usertype`='pending'"
    res=select(qry)
    return render_template("approvecouncellor.html",res1=res)
@app.route('/approved_counc',methods=['post','get'])
def approved_counc():
    qry="SELECT `counsellor`.* FROM `counsellor` JOIN `login` ON `login`.`id`=`counsellor`.`login_id` WHERE
        `login`.`usertype`='counsellor'"
    res=select(qry)
    return render_template("approvedcouncellor.html",res1=res)

@app.route('/accept_co')
def accept_co():
    id=request.args.get("id")
    qry="UPDATE `login` SET `usertype`='counsellor' WHERE `id`=%s"
    iud(qry,id)
    return redirect("/approve_counc")

@app.route('/reject_co')
def reject_co():
    id=request.args.get("id")
    qry="DELETE from `login` WHERE `id`=%s"
    iud(qry,id)
    qry="DELETE from `counsellor` WHERE `login_id`=%s"
    iud(qry,id)
    return redirect("/approve_counc")

@app.route('/managecounsellortips')
def managecounsellortips():
    qry="select * from tips where counsellor_id=%s"
    res=selectcond(qry,session['lid'])
    return render_template("c_managetips.html",val=res)

@app.route('/addcounsellortips',methods=['post'])
def addcounsellortips():
    return render_template("c_addtips.html")
@app.route('/inserttips',methods=['post'])
def inserttips():
```

Appendix

```
    tip=request.form['textfield']
    qry="insert into tips values(null,%s,%s,curdate())"
    val=(session['lid'],tip)
    iud(qry,val)
    return redirect('managecounsellortips')

@app.route('/deletetip')
def deletetip():
    id=request.args.get("id")
    qry="DELETE from `tips` WHERE `t_id`=%s"
    iud(qry,id)
    return redirect("/managecounsellortips")


app.run(debug=True)
```

dbconnect.py

```
import pymysql
def iud(q,val):
    con=pymysql.connect(host="localhost",user="root",password="",port=3306,db="intelligent chatbot")
    cmd=con.cursor()
    cmd.execute(q,val)
    id = con.insert_id()
    con.commit()
    return id
def select(q):
    con = pymysql.connect(host="localhost", user="root", password="", port=3306, db="intelligent chatbot")
    cmd = con.cursor()
    cmd.execute(q)
    s =cmd.fetchall()
    con.commit()
    return s
def selectcond(q,val):
    con = pymysql.connect(host="localhost", user="root", password="", port=3306, db="intelligent chatbot")
    cmd = con.cursor()
    cmd.execute(q,val)
    s=cmd.fetchall()
    return s
def selectone(q):
    con = pymysql.connect(host="localhost", user="root", password="", port=3306, db="intelligent chatbot")
    cmd = con.cursor()
    cmd.execute(q)
    s=cmd.fetchone()
    return s
def selectonecond(q,val):
    con = pymysql.connect(host="localhost", user="root", password="", port=3306, db="intelligent chatbot")
    cmd = con.cursor()
    cmd.execute(q,val)
    s=cmd.fetchone()
    return s
def androidselectall(q,val):
    con=pymysql.connect(host='localhost',port=3306,user='root',passwd='',db='intelligent chatbot')
    cmd=con.cursor()
    cmd.execute(q,val)
    s=cmd.fetchall()
    row_headers = [x[0] for x in cmd.description]
    json_data = []
    print(json_data)
    for result in s:
        json_data.append(dict(zip(row_headers, result)))
    return json_data
```

Appendix

```
def androidselectallnew(q):
    con=pymysql.connect(host='localhost',port=3306,user='root',passwd='',db='intelligent chatbot')
    cmd=con.cursor()
    cmd.execute(q)
    s=cmd.fetchall()
    row_headers = [x[0] for x in cmd.description]
    json_data = []
    print(json_data)
    for result in s:
        json_data.append(dict(zip(row_headers, result)))
    return json_data

    webservice.py

   import matplotlib.pyplot as plt
        import datetime

        # giving the values against
        # each value at x axis
        y = pc
        plt.bar(x, y)

        # setting x-label as pen sold
        plt.xlabel("Emotion")

        # setting y_label as price
        plt.ylabel("Count")
        plt.title(" Emotion Graph")
        fn = datetime.datetime.now().strftime("%Y%m%d%H%M%S") + ".png"
        plt.savefig("static/graph/" + fn)
        rr = fn
        plt.close()
    print(rr, "++++++++++++++++")


    return jsonify({"res":str(res),"fn":rr})



@app.route('/photoupload',methods=['post'])
def photoupload():
    file=request.files['files']
    fname=secure_filename(file.filename)
    file.save(os.path.join('static/profile',fname))
    uid=request.form['lid']
    qry="update users set image=%s where lid=%s"
    val=(fname,uid)
    iud(qry,val)
    return jsonify({"task": 'success'})




app.run(host='0.0.0.0',port=5000)
```

Database Design

Attribute Name	Datatype	Width	Description
uid	Integer	11	Primary Key
lid	int	11	Not Null
fname	Varchar	44	Null
lname	Varchar	45	Null
place	Varchar	45	Null
phone	bigint	20	Null
email	Varchar	45	Null

Table A.1: User

Attribute Name	Datatype	Width	Description
id	Integer	11	Primary Key
username	Varchar	45	Null
password	Varchar	54	Null
usertype	Varchar	45	Null

Table A.2: Login

Attribute Name	**Datatype**	**Width**	**Description**
cid	Integer	11	Primary Key
user$_l id$	int	11	Null
date	date	Null	
complaint	Varchar	44	Null
reply	Varchar	45	Null

Table A.3: Complaint

Attribute Name	**Datatype**	**Width**	**Description**
fid	Integer	11	Primary Key
user$_l id$	int	11	Null
date	date	Null	
feedback	Varchar	45	Null
reply	Varchar	45	Null

Table A.4: Feedback

DaTaflow Diagram

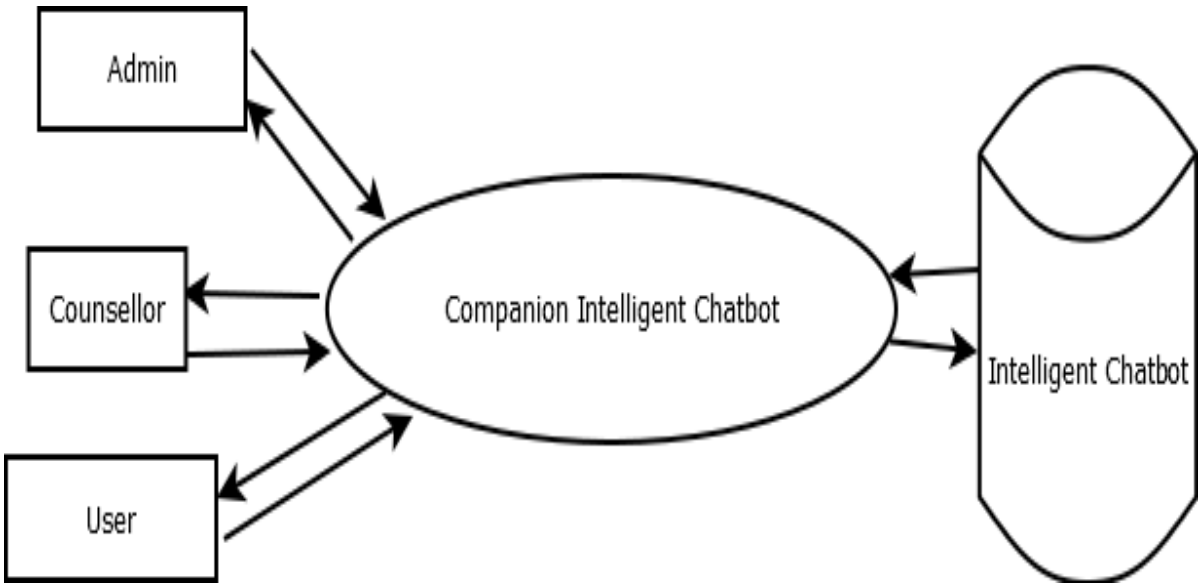


Figure A.1: Level 0

DaTaflow Diagram

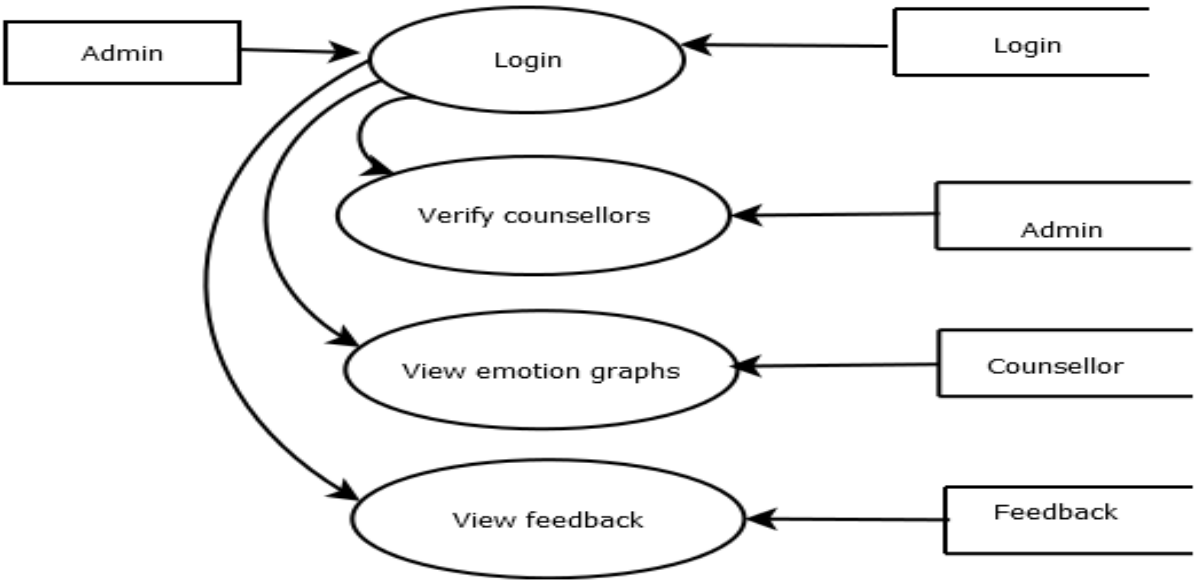


Figure A.2: Level 1.1

DaTaflow Diagram

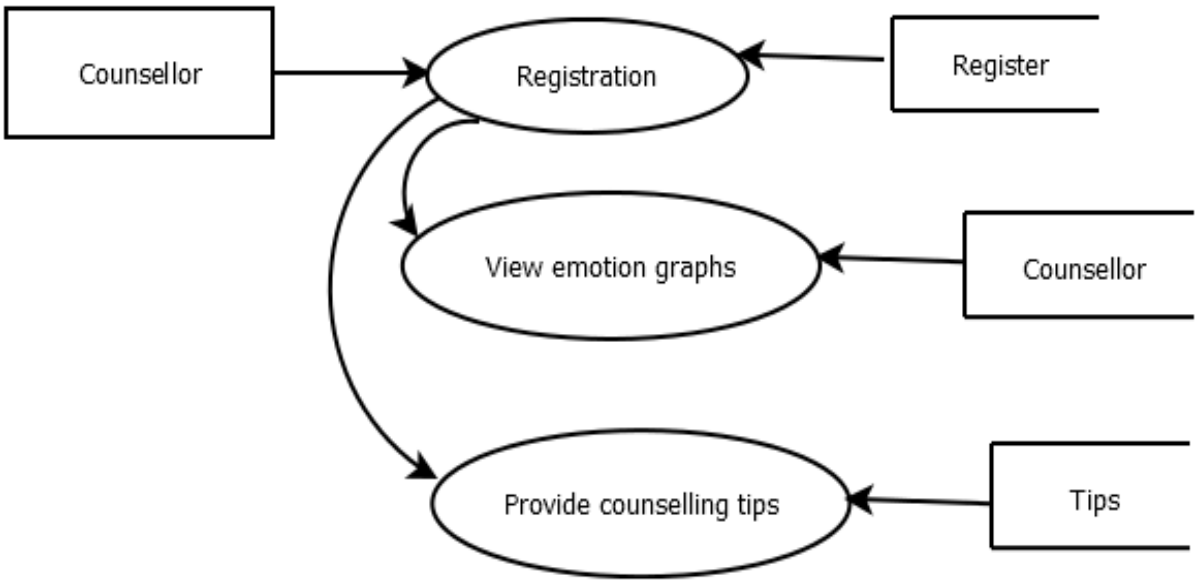


Figure A.3: Level 1.3

DaTaflow Diagram

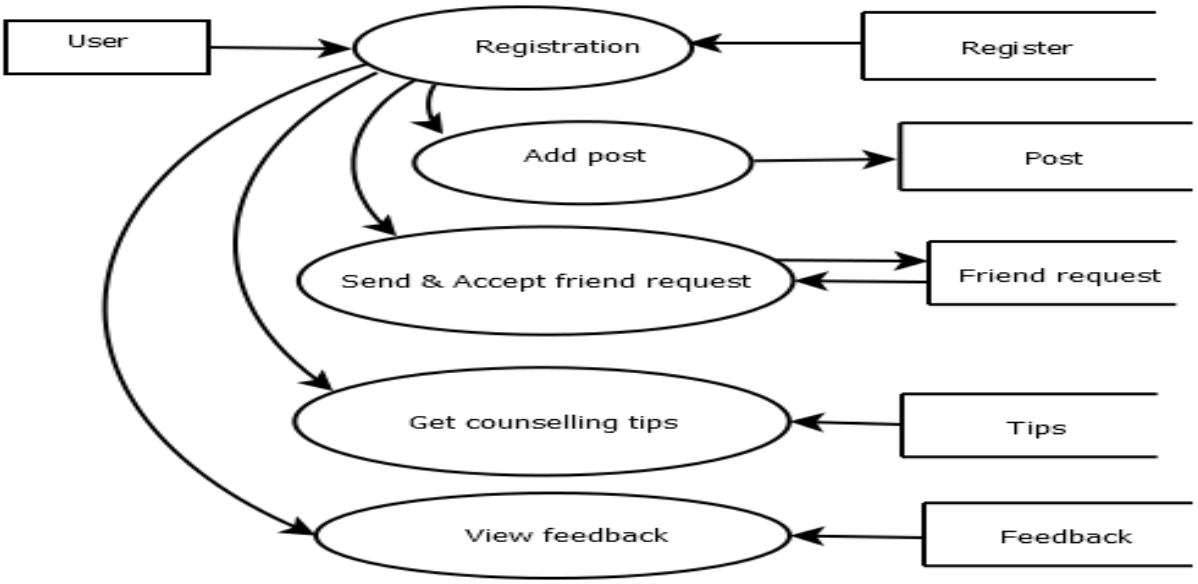


Figure A.4: Level 1.2

Screenshots

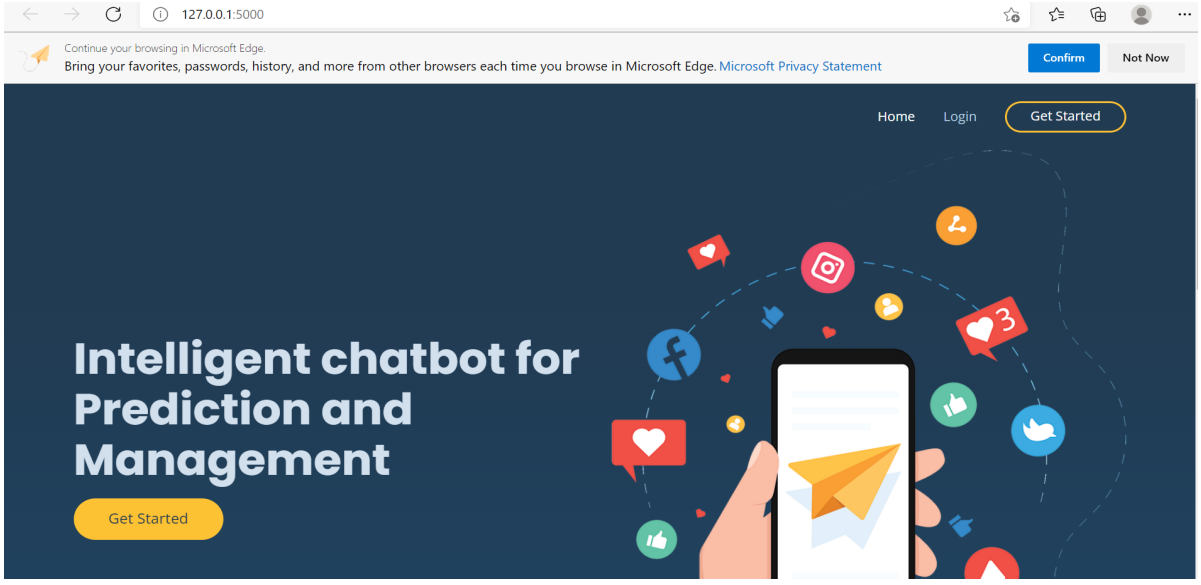


Figure A.5: Screenshot

Screenshots

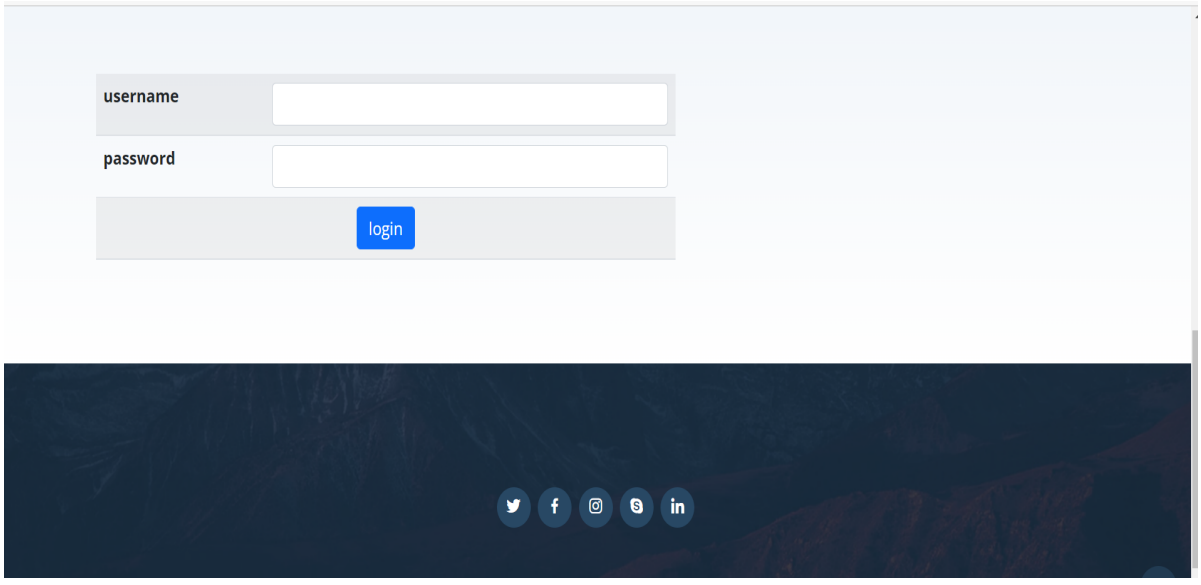


Figure A.6: Screenshot

Screenshots

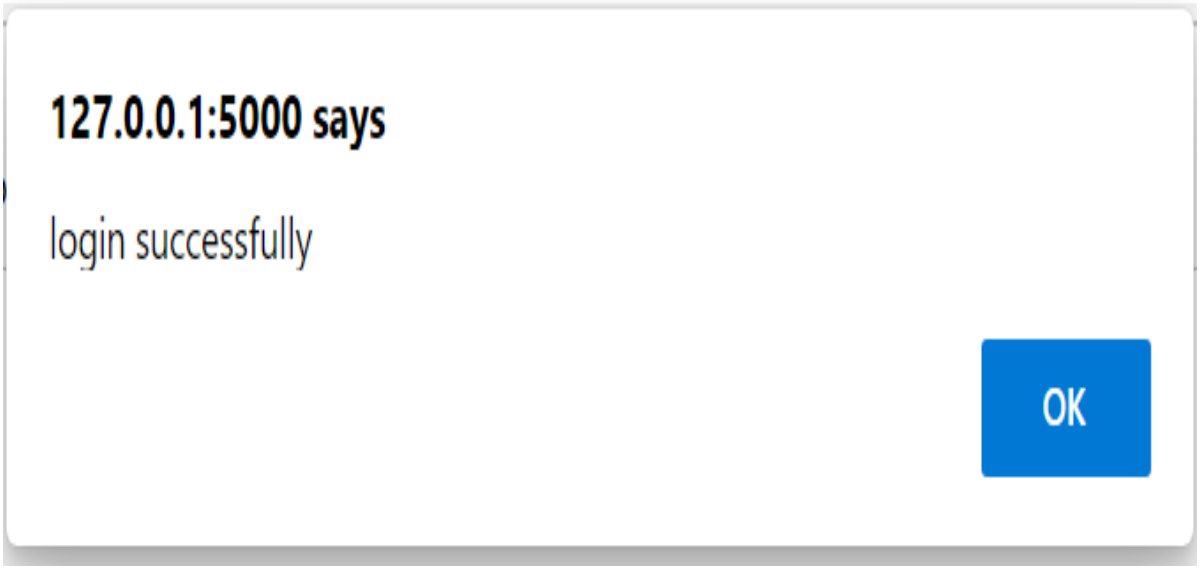


Figure A.7: Screenshot

Screenshots

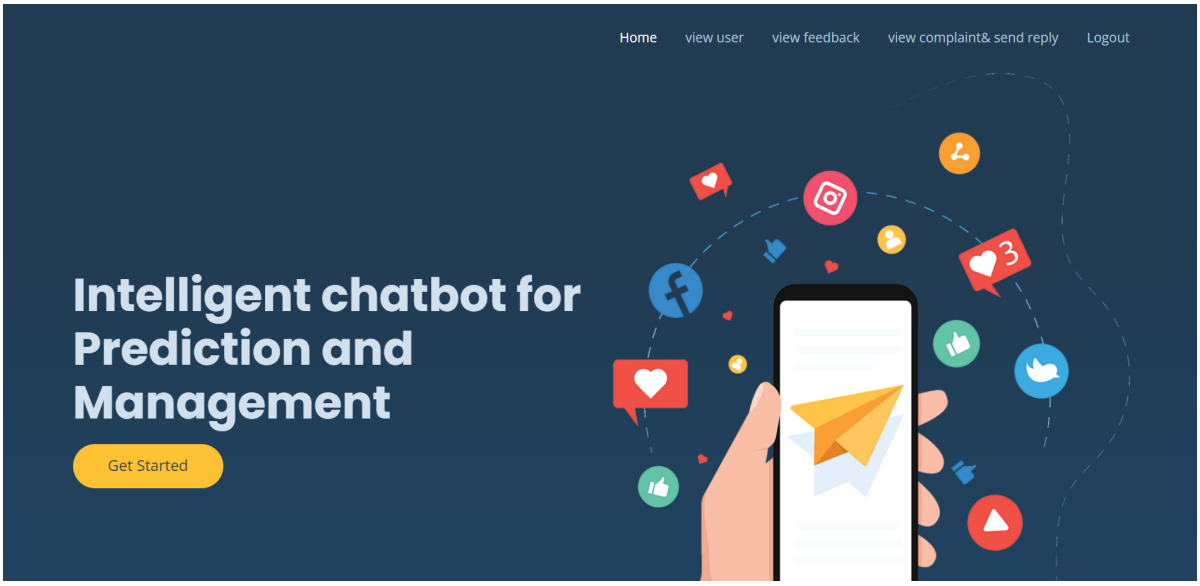
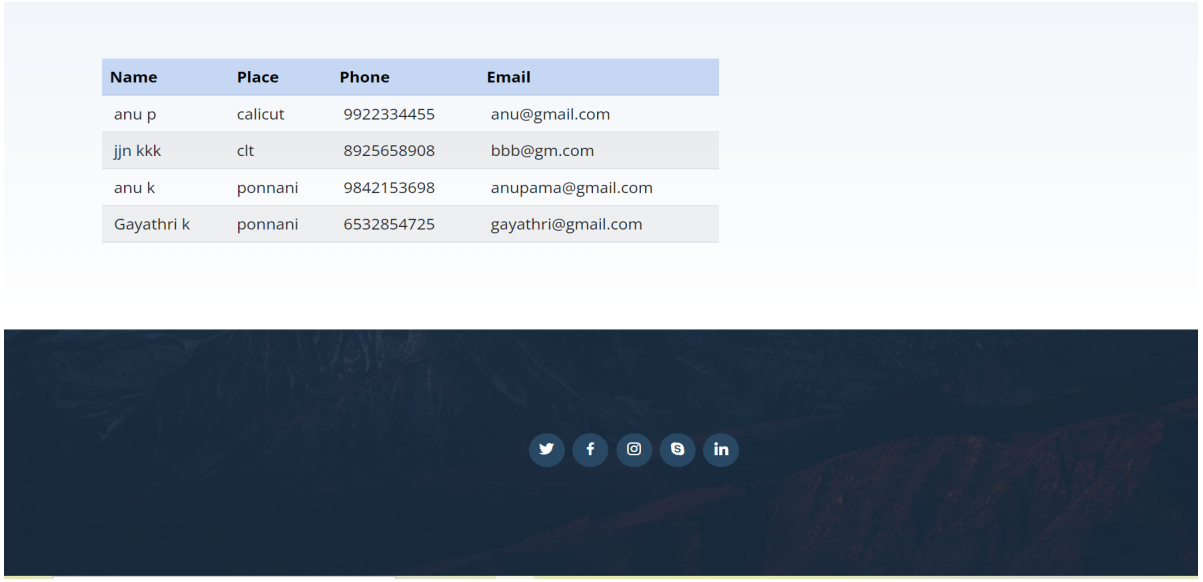


Figure A.8: Screenshot

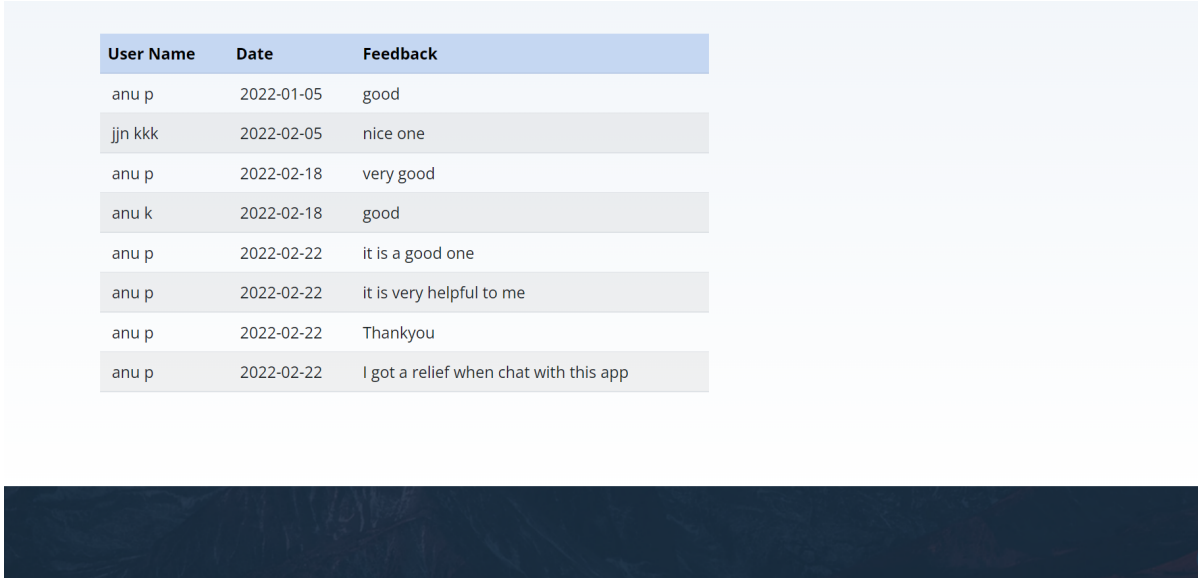
Screenshots



Name	Place	Phone	Email
anu p	calicut	9922334455	anu@gmail.com
jjn kkk	clt	8925658908	bbb@gm.com
anu k	ponnani	9842153698	anupama@gmail.com
Gayathri k	ponnani	6532854725	gayathri@gmail.com

Figure A.9: Screenshot

Screenshots



User Name	Date	Feedback
anu p	2022-01-05	good
jjn kkk	2022-02-05	nice one
anu p	2022-02-18	very good
anu k	2022-02-18	good
anu p	2022-02-22	it is a good one
anu p	2022-02-22	it is very helpful to me
anu p	2022-02-22	Thankyou
anu p	2022-02-22	I got a relief when chat with this app

Figure A.10: Screenshot

Screenshots

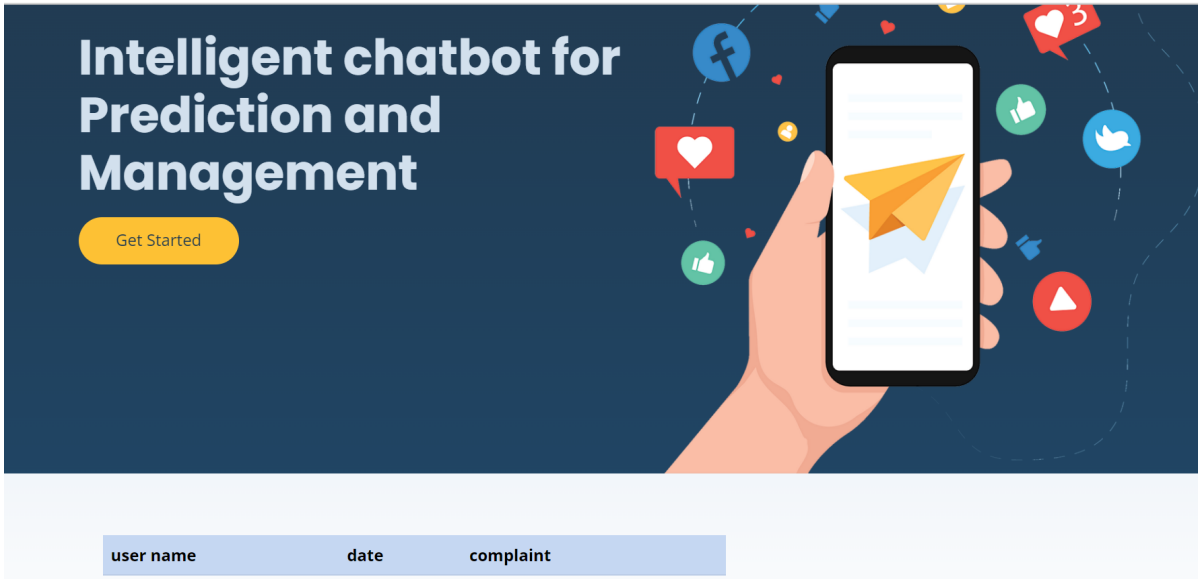


Figure A.11: Screenshot

Screenshots

Figure A.12: Screenshot

Screenshots

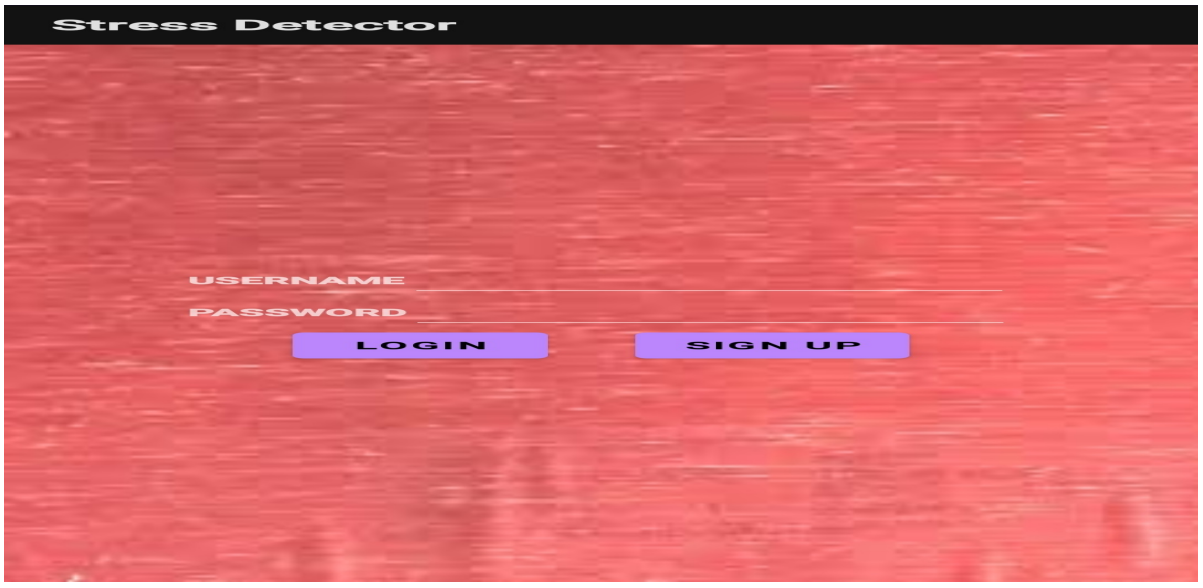


Figure A.13: Screenshot

Screenshots

Figure A.14: Screenshot

Screenshots

Figure A.15: Screenshot

Screenshots

Figure A.16: Screenshot

Screenshots

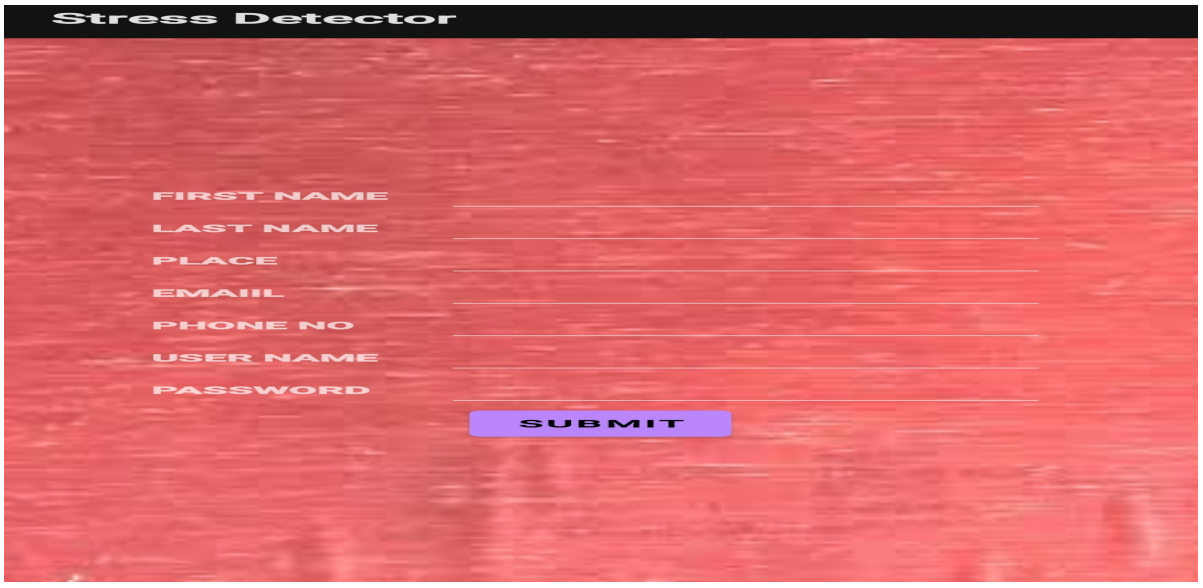


Figure A.17: Screenshot

Screenshots

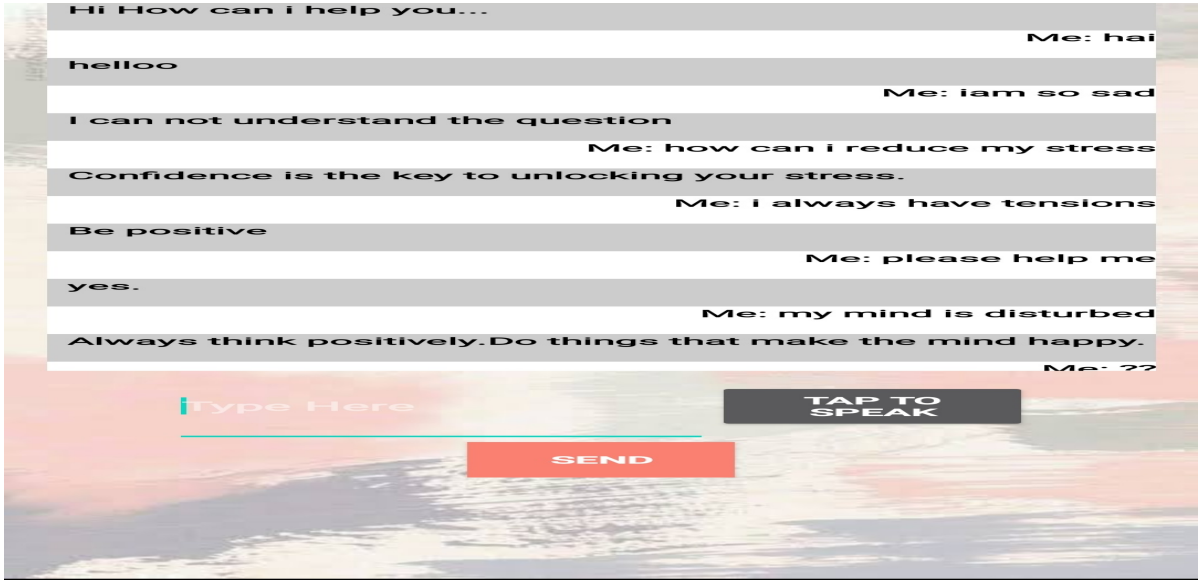


Figure A.18: Screenshot

Screenshots

Figure A.19: Screenshot