

ONLINE DEPRESSION DETECTION IN HUMANBEINGS

*A Project Report Submitted
in partial fulfillment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

AWARDED BY

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY
KAKINADA**

BY

MEDIDI TIRU VENKATA PHANI SOWMYA (17H41A0588)

KUMPATLA SOWMYA HARINI DEVI (17H41A0583)

SUNKARA SAI TEJA PRASAD (18H45A0508)

NAMANA VENKATESH (17H41A0593)

Under the Esteemed guidance of

MR.B.V.RAMKUMAR-M.tech,MISTE,MCSI,(Ph.D)

Professor

Department of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAM VENKATA CHALAMAYYA INSTITUTE OF TECHNOLOGY AND SCIENCE

**(Approved by AICTE, Accredited by NAAC, Permanently Affiliated to JNTUK,
ISO 9001:2015 QMS Certified Institution)**

Amalapuram-533 201

2017-2021

BONAM VENKATA CHALAMAYYA INSTITUTE OF TECHNOLOGY AND SCIENCE

(Approved by AICTE, Accredited by NAAC, Permanently Affiliated to JNTUK,

ISO 9001:2015 QMS Certified Institution)

Amalapuram-533 201



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION BY THE CANDIDATES

We Ms. **Tiru Venkata Phani Sowmya, Kumpatla Sowmya Harini Devi, Sunkara Sai Teja Prasad, Namana Venkatesh** bearing hall ticket numbers **17H41A0588, 17H41A0583, 18H45A0508, 17H41A0593** hereby declare that the main project report entitled “**ONLINE DETECTION DEPRESSION IN HUMANBEINGS** ”, under the esteemed guidance of **MR.B.V.RAM KUMAR- M.Tech,MISTE,MCSI,(Ph.D)** **Professor, Department of CSE**, is submitted in partial fulfillment of the requirements for the award of the degree of bachelor of technology in Computer Science Engineering.

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

MEDIDI TIRU VENKATA PHANI SOWMYA (17H41A0588)

KUMPATLA SOWMYA HARINI DEVI (17H41A0583)

SUNKARA SAI TEJA PRASAD (18H45A0508)

NAMANA VENKATESH (17H41A0593)

**BONAM VENKATA CHALAMAYYA INSTITUTE OF TECHNOLOGY
AND SCIENCE**

**(Approved by AICTE, Accredited by NAAC, Permanently Affiliated to JNTUK,
ISO 9001:2015 QMS Certified Institution)**

Amalapuram-533 201



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

*This is to certify that the main project work “**ONLINE DEPRESSION DETECTION IN HUMANBEINGS**” submitted by **Medidi Tiru Venkata Phani Sowmya, Kumpatla Sowmya Harini Devi, Sunkara Sai Teja Prasad, Namana Venkatesh** is examined and adjudged as sufficient as a partial requirement for the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** at Jawaharlal Nehru Technological University, Kakinada is a bonafide record of the work done by these group of students under my guidance and supervision.*

PROJECT GUIDE

Mr. B.V. Ram Kumar

M.Tech,MISTE,MCSI,(Ph.D)

Associate Professor

DEPARTMENT OF C S E

HEAD OF THE DEPARTMENT

Mr. B. V. Ram Kumar

M.Tech,MISTE,MCSI,(Ph.D)

Associate Professor

DEPARTMENT OF C S E

PROJECT EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would like to express my sincere gratitude and indebtedness to my project supervisor **Mr. B. V. RAM KUMAR** M.Tech, MISTE, MCSI, (Ph.D) Professor, CSE Department for his valuable suggestions and interest throughout the course of this project.

We also thankful to Head of the Department **Mr. B. V. RAM KUMAR** M.Tech, MISTE, MCSI, (Ph.D) for guiding us for completing this project successfully.

We also thankful to Head of the Institution **Dr.G.M.V.PRASAD** B.E, M.Tech, Ph.D, FIETE, FIE, MIEEE, MSEMCE, MISTE for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family members for their support throughout our carrier.

Medidi Tiru Venkata Phani Sowmya	(17H41A0588)
Kumpatla Sowmya Harini Devi	(17H41A0583)
Sunkara Sai Teja Prasad	(18H45A0508)
Namana Venkatesh	(17H41A0593)

DEPARTMENT OF COMPUTER & SCIENCE ENGINEERING (CSE)

VISION

To become a prominent department of Computer Science & Engineering producing competent professionals with research and innovation skills, inculcating moral values and societal concerns.

MISSION

- To provide a learning ambience to enhance innovations, problem solving skills, leadership qualities, team spirit ,ethical and social responsibilities.
- Establish Industry Institute Interaction program to enhance latest technologies and the entrepreneurship skills.
- Conduct research to advance the state of the art in computer science & integrate research results and innovations into other scientific disciplines.
- To empower the students with required skills to solve the complex technological problems of modern society.

PROGRAM EDUCATIONAL OBJECTIVES (PEO'S):

PEO 1: Graduates will be an efficient software developer in diverse fields and will be a successful professional and/or pursue higher studies.

PEO 2: Graduates will be capable to adapt to new computing technology for professional excellence and research and be a lifelong learner.

PEO 3: Graduates will work productively exhibiting ethical qualities for the betterment of society.

PEO 4: Graduates will possess leadership qualities, work harmoniously as a team member with effective communication skills.

PROGRAM SPECIFIC OBJECTIVES (PSO)

PSO 1: Gain capability to use current techniques, skills & tools necessary for carrying out multidisciplinary projects.

PSO 2: Achieve an ability to build a computer based system, process or a component that meets the desired needs.

PSO 3: Acquaint with the contemporary trends in industrial/research setting and thereby innovate novel solutions to existing problems.

ABSTRACT

- In this project we are detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them. Now-a-days peoples are using online post services to interact with each other compare to human to human interaction. So by analysing users post this application can detect depression and send motivation messages to them. Administrator of this application will send motivation messages to all peoples who are in depression.
- To detect depression we are using SVM (support vector machine) algorithm which analyse users post and give result as negative or positive. If users express depression words in post then SVM detect it as a negative post else positive post.

TABLE OF CONTENTS

1. INTRODUCTION	1-3
1.1 About the Project	
1.2 Objective of the project	
1.3 Purpose and Scope	
1.4 Existing system	
1.5 Proposed system	
2. LITERATURE SURVEY	4-7
3. SYSTEM ANALYSIS	8-17
3.1 Feasibility Study	
3.1.1 Economical Feasibility	
3.1.2 Technical Feasibility	
3.1.3 Social Feasibility	
3.2 Requirement Specification	
3.2.1 Functional Requirements	
3.2.2 Non -Functional Requirements	
3.3 System Specification	
3.3.1 Hardware Requirements	
3.3.2 Software Requirements	
4. SYSTEM DESIGN	18-29
4.1 System Design	
4.1.1 System Modules	
4.1.2 System Architecture	
4.2 UML Diagrams	
5. TECHNOLOGY	30-41
6. CODING	42-53
7. TESTING	54-57
8. OUTPUT SCREENS	58-65
9. USER MANUAL	65-70
10. CONCLUSION AND FUTURE ENHANCEMENT	71-72
11. BIBLIOGRAPHY	73-75

ONLINE DETECTION DEPRESSION IN HUMAN BEINGS

CHAPTER -1

INTRODUCTION

The chapter gives brief introduction of the project

1.1 About the Project:

- The proliferations of internet and communication technologies, especially the online social networks have rejuvenated how people interact and communicate with each other electronically. The applications such as Facebook, Twitter, Instagram and alike not only host the written and multimedia contents but also offer their users to express their feelings, emotions and sentiments about a topic, subject or an issue online. On one hand, this is great for users of social networking site to openly and freely contribute and respond to any topic online other hand, it creates opportunities for people working in the health sector to get insight of what might be happening at mental state of someone who reacted to a topic in a specific manner.
- In order to provide such insight, machine learning techniques could potentially offer some unique features that can assist in examining the unique patterns hidden in online communication and process them to reveal the mental state (such as 'happiness', 'sadness', 'anger', 'anxiety', depression) among social networks' users. Moreover, there is growing body of literature addressing the role of social networks on the structure of social relationships such as breakup relationship, mental illness ('depression', 'anxiety', 'bipolar' etc.), smoking and drinking relapse, sexual harassment and for suicide ideation.
- Therefore the main aim of this project is to help the depressed people to come out of depression through notifying them with the motivated posts and also the data that is uploaded by the depressed people in the app helps the diagnostics to treat in a better way by knowing their feelings.

1.2 OBJECTIVE OF THE PROJECT

- In this project we are detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them. Now-a-days peoples are using online post services to interact with each other compare to human to human interaction. So by analysing users post this application can detect depression and send motivation messages to them. Administrator of this application will send motivation messages to all peoples who are in depression. To detect depression we are using SVM (support vector machine) algorithm which analyse users post and give result as negative or positive. If users express depression words in post then SVM detect it as a negative post else positive post.

1.3 PURPOSE AND SCOPE

PURPOSE:

- The purpose of this project is to develop an application for detecting depression based on the posts uploaded by the users. Detection Depression is nothing but the act of trying to determine whether the person is depressed or not based on the posts he or she uploaded. That means the posts are either image or audio or text files. The hope is that with a greater understanding of how the classifier can detect the depression based on the posts uploaded, from that the admin can identify whether the person is depressed or not. Requirements of the detection are the images, audios and text features from the user. Social networks have been developed as a great point for its users to communicate with their interested friends and share their opinions, photos, and videos reflecting their moods, feelings and sentiments. This creates an opportunity to analyze social network data for user's feelings and sentiments to investigate their moods and attitudes when they are communicating via these online tools.

Scope:

- The User can upload any of the three options that are images, audio files and text files based on that the admin can able to identify whether the person is depressed or not. For instance the user randomly uploads a post then based on that he gets a motivated post if the output we get is positive. Here positive means the user is depressed and negative means the user is not depressed. This project will exclusively focus on detecting the depression through online posts.

1.4 EXISTING SYSTEM:

- Now-a-days peoples are using online post services to interact with each other compare to human to human interaction.
- we aim to analyze Facebook data to detect any factors that may reflect the depression of relevant Facebook's users. Various machine learning techniques are employed for such purpose.
- Considering the key objective of this study, the following are subsequent research challenges addressed in paper.

Problem Analysis:

- As users express their feeling as a post or comments in the Face-book platform, sometimes their posts and comments refer to as emotional state such as ‘joy’, ‘sadness’, ‘fear’, ‘anger’, or ‘surprise’.
- Less Security.
- It does not give accurate results.
- We are doing a survey on ocr techniques

1.5 PROPOSED SYSTEM:

- Detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them.
- To implement this project we are using python Speech Recognition API which will read text from audio files and then SVM will analyse that text to detect depression, user can also upload images via post and python Tesseract OCR (Optical Character Recognition) API can read text from uploaded image and then SVM will detect depression from that text, User can upload post in text file also.
- Administrator of this application will send motivation messages to all peoples who are in depression. To detect depression we are using SVM (support vector machine) algorithm which analyse users post and give result as negative or positive. If users express depression words in post then SVM detect it as a negative post else positive post.
- **SUPPORT VECTOR MACHINE:**

A support vector machine is machine learning algorithm that analyzes data for classification and regression analysis .It is a supervised machine learning method that looks at data and sorts it into one of the two categories .They are used in text categorization, image classification ,handwriting recognition etc.

ADVANTAGES:

- Security.
- Gives better accuracy compared to Existing System.
- Sends Motivation posts to the users who are in Depression.

CHAPTER -2

LITERATURE SURVEY

It gives us the Summary of the Previous Researches on the project

- There is growing body of literature that analyses the properties of depression. Choudhury et al. argue that depression constitutes a genuine test in individual and general wellbeing. Considerable number of individuals experiences the ill-effects of despondency and just a division gets sufficient treatment every year. They also investigated the possibility to utilize online networking to identify and analyze any sign of significant depression issue in people. Through their web-based social
- networking postings, they quantified behavioral credits identifying with social engagement, feeling, dialect and semantic styles, sense of the self-system, and notices of antidepressant medications.
- Choudhury et al. considered online networking as a promising instrument for public health, concentrating on the utilization of Twitter presents on fabricating predictive models about the forthcoming impact of childbirth on the conduct and disposition of new mothers. Utilizing Twitter posts, they measured postpartum changes in 376 mothers along measurements of social engagement, feeling, informal community, and phonetic style. O'Dea et al. examined that Twitter is progressively researched as methods for recognizing psychological well-being status, including depression and suicidality in the population. Their investigation revealed that it is conceivable to recognize the level of worry among suicide-related tweets, utilizing both human coders and a programmed machine classifier.
- Zhang et al. have shown that if individuals with a high danger of suicide can be recognized through online networking like microblog, it is conceivable to actualize a dynamic intervention system to save their lives.
- Many researchers have demonstrated that utilizing user-created content (UGC) accurately may help decide individuals' psychological wellness levels. For instance, Aldarwish and Ahmad examined that the utilization of Social Network Sites (SNS) is expanding these days, particularly by the more youthful eras. Because the accessibility of SNS enables clients to express their interests, sentiments and offer day by day schedule.
- Nguyen et al. utilized machine learning and statistical strategies to separate online messages amongst depression and control groups utilizing temperament, psycholinguistic procedures and substance subjects removed from the posts created by individuals from these groups.
- Park et al. investigated states of mind and practices toward online web-based social networking in view of whether one is discouraged or not. They directed semi-organized up close and personal meetings with 14 dynamic Twitter users, half of whom were discouraged and the other half non-discouraged.

- Other than they examined a few plan implications for future social networks that could better suit users with depression and give bits of knowledge towards helping discouraged users address their issues through online web-based social networking .
- Bachrach et al. studied how user's activity on Facebook identifies with their identity, as measured by the standard Five Factor Model. They analyzed relationships between user's identity and the properties of their Facebook profiles.
- For instance, the size and thickness of their friendship network, number of transferred photographs, and number of occasions went to, number of gathering enrolment's, and number of times the user has been tagged in photographs. Ortigosa et al. have exhibited a new strategy for sentiment examine in Facebook that suggests that starting from messages composed by users, as to extract data about the users' assessment extremity (positive, unbiased or negative), as transmitted in the messages they write; and to show the users' standard conclusion extremity and to distinguish huge passionate changes.
- In the context of Facebook mining, Holleran found initial evidence that depression is a major contributor to the overall global burden of diseases. In other related work, Wang et al. and Shen et al. examined various depression-related features, and built a multimodal depressive model to detect the depressed users.
- Although, some of the above reported work has discussed emotional process, temporal process, linguistic style to detect depression, the following shortcomings are observed in the existing literature:
- There are few individual studies that have applied SVM, KNN, Decision Tree and Ensemble separately. There are no well-known studies that have combined all these techniques together at same dataset to investigate the variations in technique-based findings.
- There is no significant study that has applied the above-mentioned machine learning techniques on Facebook data for depression detection.
- To address the above-listed shortcomings, we make an attempt to detect depression from Facebook comments with the present work; we expand the scope of social media-based depression measures, describing the different features of Facebook user comments.
- We applied machine learning approaches that can use those measures for the detection of individuals who are suffering with depression.

CHAPTER - 3

SYSTEM ANALYSIS

This chapter gives the details of the system and data design.

3.1 FEASIBILITY STUDY

- An analysis of the ability to complete a project successfully, taking into account legal, economic, technological, scheduling. Rather than just diving into a project and hoping for the best, a feasibility study allows project managers to investigate the possible negative and positive outcomes of a project.
 - A feasibility study evaluates the project's potential for success.
 - Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing project.
- Many Feasibility studies are disillusioning for both users and analysts. First, the study often presupposes that when feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to over look the confusion inherent in the system development the constraints and the assume attitudes. The feasibility study is to serve as a decision document for a project. Three key considerations involved in the feasibility analysis:
 - Economical Feasibility
 - Technical Feasibility
 - Social Feasibility

3.1.1 ECONOMICAL FEASIBILITY:

The purpose of economic feasibility study is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency.

The project “ONLINE DEPRESSION DETECTION IN HUMAN BEINGS” is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.

3.1.2 TECHNICAL FEASIBILITY:

Technical feasibility is concerned with specifying equipment and software that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include the facility to produce outputs in a given time, response time under certain conditions.

The project “ONLINE DETECTION DEPRESSION IN HUMANBEINGS” is technically feasible as the system developed has high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.1.3 SOCIAL FEASIBILITY:

The affect that a proposed project may have on the social system in the project environment is addressed in the social feasibility. It may happen that a particular category of employees may be short or not available as a result of ambient social structure. The influence on the social status of the participants by the project should be evaluated in order to guarantee compatibility. It must be identified that employees in the particular industries may have specific status symbols with in the society. The aspect of study is to check the level of acceptance of the system by the user.

The project “ONLINE DEPRESSION DETECTION IN HUMAN BEINGS” includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 REQUIREMENT SPECIFICATION:

This section provides the details of the project's need based survey, system requirements, Hardware Requirements, Software Requirements, and System Requirements.

A Software Requirements Specification (SRS) – A requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stake holders to help identify business problems and propose solutions. Within the systems development life cycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- **Business requirements** describe in business terms what must be delivered or accomplished to provide value.
- **Product requirements** describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- **Process requirements** describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development style.

A systems engineering, a requirement can be a description of what a system must do, referred to as a functional requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Non-functional requirements, or 'performance requirements' or 'quality of service requirements. Examples of such requirements include usability, availability, reliability, supportability, testability and maintainability.

A collection of requirements define the characteristics or features of the desired system. A 'good' list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called "implementation bias" or "solution engineering". However, implementation constraints on the solution may validly be expressed by the future owner, for example for required interfaces to external systems; for interoperability with other systems; and for commonality (e.g. of user interfaces) with other owned products. In Software Engineering, the same meanings of requirements apply, except that the focus of interest is the software itself.

3.2.1 Functional requirements:

- In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.
 - Functional software requirements help you to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform.
-
- Collection of different images, text and audio files will be treated as inbuilt dataset.
 - Gives result either positive or negative based on the posts uploaded by the user.
 - A graph is generated from the users post which indicates the rate of depressed and non depressed people.
 - Based on the result if outcome is positive then the user is said to be depressed and a motivated post is sent them through the admin side.

3.2.2 Non –Functional Requirements:

- A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?
- A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.
- Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are “constraints”, “quality attributes”, “quality goals”, “quality of service requirements” and “non-behavioral requirements”.

Accessibility:

Accessibility is a general term used to describe the degree to which a product, device or service is available to as many people as possible. Accessibility can be viewed as the “ability to access” and benefit from some system or entity. Accessibility is often used to focus on people with disabilities or special needs and their right of access to entities, often to use of assistive technology. This project is accessible for understanding of ,how we detect depression from the users online data.

We are improving confidence in predictions, particularly in audio and text files as result of improved understanding of speech based depression recognition.

Availability:

Availability is the degree to which a system, sub-system, or equipment is in a specified operable and committable state at the start of a mission, when a mission is called for at an unknown, i.e., a random, time. Simply put, availability is the proportion of time a system is in a functioning condition. This is often described as a mission capable rate; it is available in this project because equipments used in this one are operable.

Scalability:

- Scalability is the ability of a system, network, or process, to handle growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth. For example, it can refer to the capability of a system to increase total throughput under an increased load when recourses (typically hardware) are added. An analogous meaning is implied when the word is used in a commercial context, where scalability of a company implies that the underlying business model offers the potential for economic growth with the company.

- “ONLINE DEPRESSION DETECTION IN HUMAN BEINGS”models in order to emphasize the close liaison between the applications and the hardware used to run these models moreover,the gender prediction tries to provide the ways to turn the applications to achieve the best possible speed up and best utilization of the processing involved.

Portability:

- Portability in high-level computer programming is the usability of the same software in different environments. The pre requirement for portability is the generalized abstraction between the application logic and system interfaces. It is portable because it runs on different environment.
- Detection of Depression using classifiers in machine learning contributes by exploring the use of various data mining techniques for depression detection Techniques include: Support Vector Machine (SVM), k Nearest Neighbor (kNN), Decision Tree (J48) and Random Forest. The dataset is obtained from google and consists of several gender related attributes. For effective prediction, pre-processing technique is used which consists of cleaning and normalization processes. Performance of used data mining techniques is analyzed in terms of precision, recall and f-measure with various ratios of training and test data.

Robustness:

Robustness is the ability of a computer system to cope with errors during execution or the ability of an algorithm to continue to operate despite abnormalities in input ,calculations,etc.Formal techniques,such as fuzz testing, or essential to showing robustness since this type of testing involves invalid or unexpected inputs. Various commercial products perform robustness testing of software systems.

Performance:

- Performance defines how fast a software system or its particular piece responds to certain users’ actions under certain workload.

- In most cases, this metric explains how much a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it's not always like that. Performance requirements may describe background processes invisible to users, e.g. backup. But let's focus on user-centric performance.
- “ONLINE DEPRESSION DETECTION IN HUMAN BEINGS” models in order to emphasize the close liaison between the applications and the hardware used to run these models moreover ,the depression detection tries to provide the ways to turn the applications to achieve the best possible speed up and best utilization of the processing involved. It can gives the accurate performance with effective accuracy score and prediction of gender with increase number of images.

3.3 SYSTEM SPECIFICATIONS:

3.3.1 Hardware Requirements :

- Processor : Intel Pentium IV OR Higher
- RAM : minimum 1 GB
- Secondary Storage : minimum 160 GB
- Key Board : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse

3.3.2 Software Requirements:

- Operating System : Windows XP/Windows 7/8/10
- Scripting languages : Python 3.6
- Python Packages : Matplotlib,Pandas
- For developing the Application
 1. Python
 2. Django
 3. Mysql
 4. Mysqlclient
 5. WampServer 2.4

CHAPTER – 4

SYSTEM DESIGN

This chapter gives the details of the system and data design.

4.1 SYSTEM DESIGN:

4.1.1 SYSTEM MODULES:

- **Admin Module:** Administrator will login to application using username as 'admin' and password 'admin'. After login admin can view all registered users and all posts send by each users. Admin can send motivation messages to all depressed users. All positive and negative depression users can also be seen in the form of graph.
- **User Module:** Users need to register with the application and then login to application to access various sub modules such as
 - **Search Friends:** Using this module user can see all peoples register with the application
 - **Upload Posts:** Using this module user can upload post in various formats such as text file, image or audio file. This application accepts only .WAV file format.
 - **View Motivation Messages:** Using this module users can view all motivation messages send by administrator.
- **Preprocessing:**
 - It is a process of transforming the raw, complex data into systematic understandable knowledge. It involves the process of finding out missing and redundant data in the dataset.
 - For Preprocessing the data uploaded in the application we use sklearn library in Python.
 - Pandas is used to read the dataset i.e for identifying the given sentence/text.
 - CNN is used to identify both image and audio posts.
- **Classification:**
 - Classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.
 - Machine Learning algorithm that is SVM is used to get the output i.e either positive or output.

- **Training:**

- Regression is basically a statistical approach to find the relationship between variables. In machine learning, this is used to predict the outcome of an event based on the relationship between variables obtained from the data-set. Linear regression is one type regression used in Machine Learning.

- **Output:**

- Output can be predicted by using Machine Learning algorithms.

4.1.2 SYSTEM ARCHITECTURE:

- A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. System architecture conveys the informational content of the elements consisting of a system, the relationships among those elements, and the rules governing those relationships.
- Data is collected and stored in NoSQL / SQL format. That data is divided into two parts
 - Training data
 - Testing data.
- Training data is used for training the model and then that model is tested using testing data. After this, the trained model is used for predicting gender of a person in given feature set.

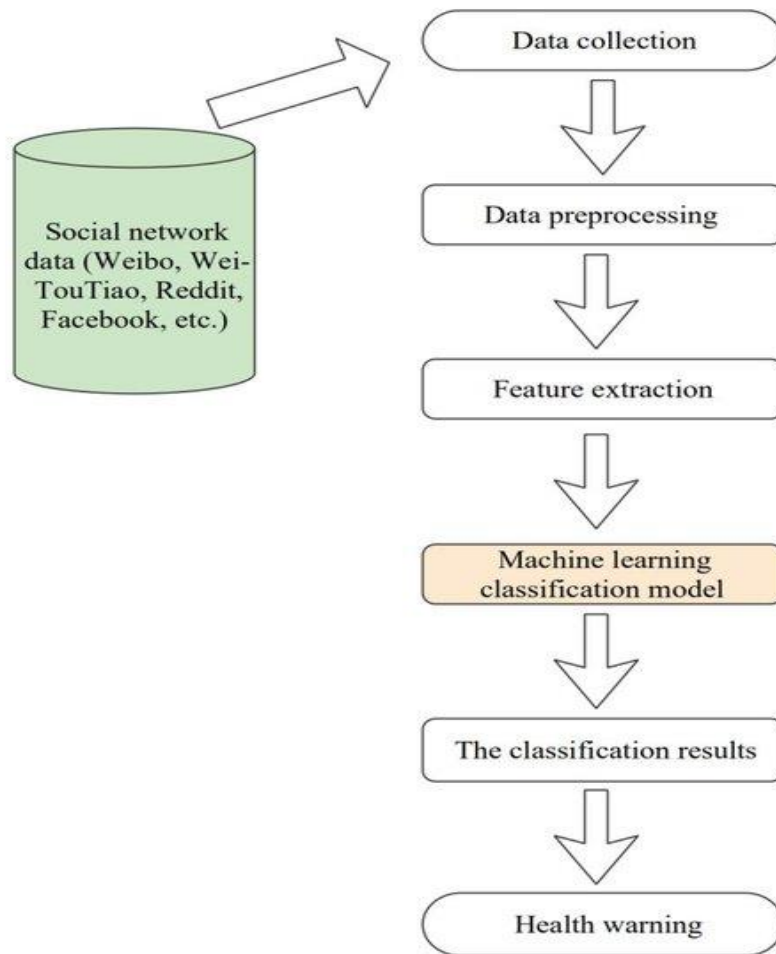


FIG: ARCHITECTURE OF DETECTION DEPRESSION USING MACHINE LEARNING

- Analyzing from the existing system we use the SVM classification model for predicting the output. Here the output we obtain is either positive or negative.
- If we get positive as outcome the administrator of the application sends the motivational Posts.
- For positive or negative outcome we make use of nltk-natural language toolkit.
- And within in the application a graph is also maintained which indicates depressed people and non depressed people for this purpose we use Matplotlib.

4.2 UML DIAGRAMS:

- To design the project, we use the UML diagrams. The Unified Modelling Language (UML) is a general- purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
- The main aim of UML is to define a standard way to **visualize** the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.
- UML diagrams to portray the **behavior and structure** of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis.
- UML is linked with **object oriented** design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:
 1. **Structural Diagrams** – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.
 2. **Behavior Diagrams** – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

GOALS OF UML:

There are a number of goals for developing UML but the most important is to define some general purpose modeling language, which all modelers can use and it also needs to be made simple to understand and use. The system can be a software or non-software system.

The primary goals in the design of the UML were:



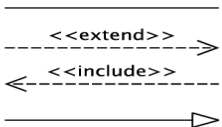
- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.

- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

USECASE DIGRAM:

- A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

- **SYMBOLS:**

Symbol	Reference Name
	Actor
	Use case
	Relationship

- They provide a good high level analysis from outside the system. Use case diagrams specify how the system interacts with actors without worrying about the details of how that functionality is implemented.

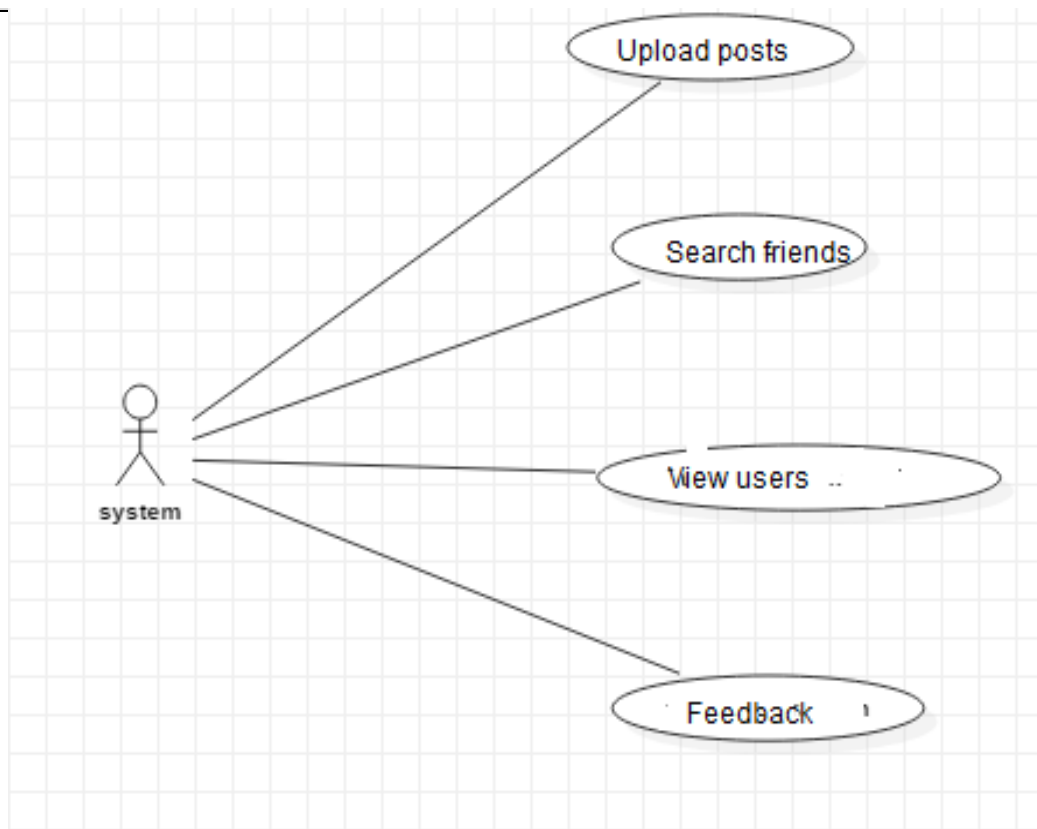


FIG:USECASE DIAGRAM

The above diagram represents the main two **actors** in the project, they are

- User
- Admin

CLASS DIAGRAM:

- It is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the objects.
- The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code.
- Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

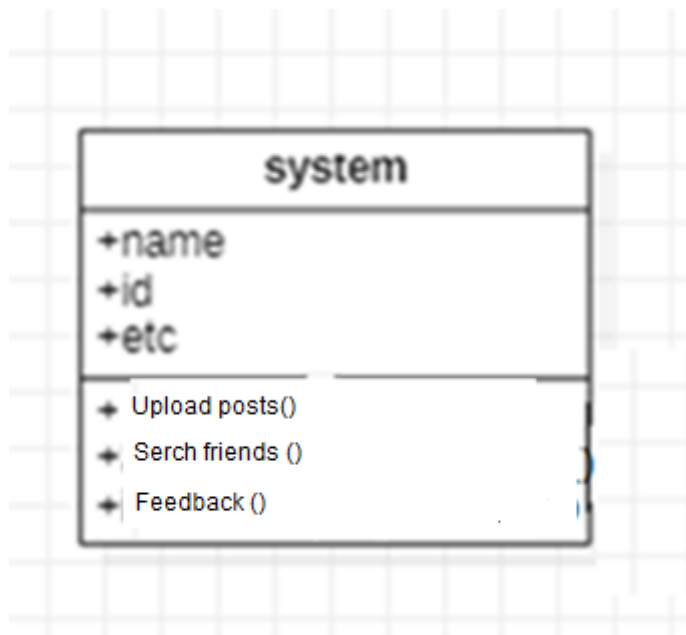


FIG :CLASS DIAGRAM

The above mentioned class diagram represents the Chatbot system workflow model. This diagram has class models with class names as

- User
- Admin
- Home screen.

SEQUENCE DIAGRAM:

- A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.
- A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

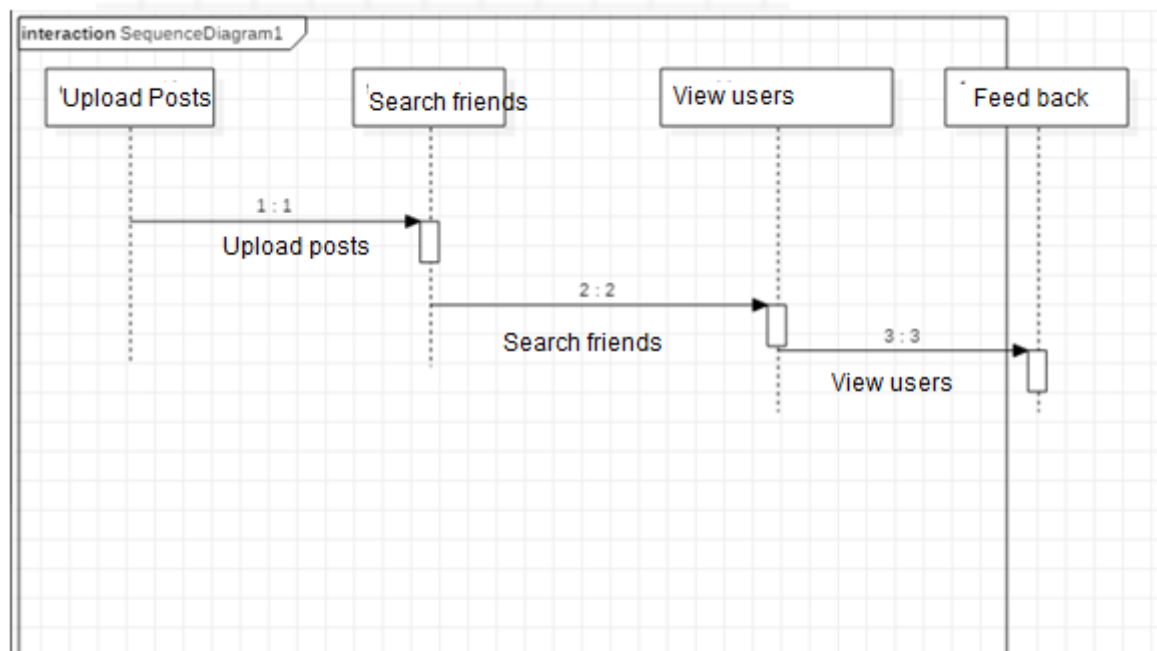


FIG:SEQUENCE DIAGRAM

The above diagram represents the sequence of flow of actions in the system.

ACTIVITY DIAGRAM:

- An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system.
- The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

- Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. It does not show any message flow from one activity to another.
- Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

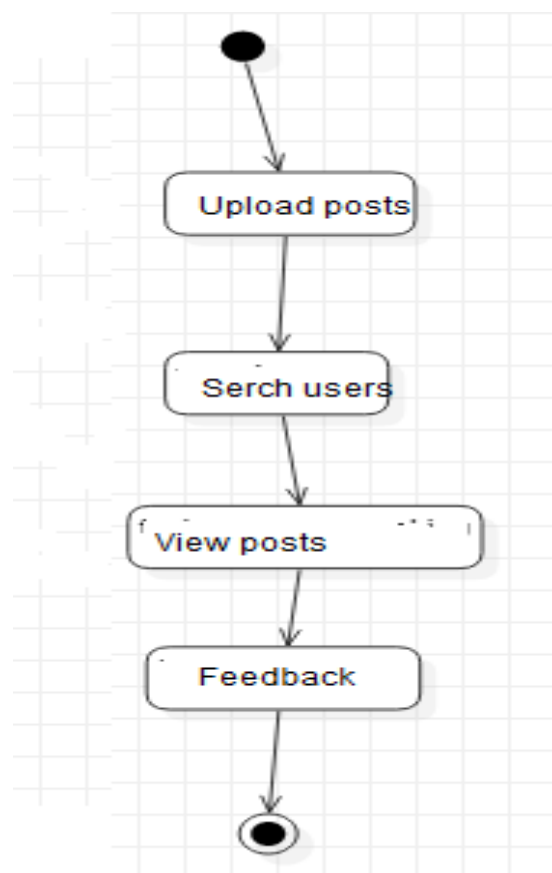


FIG:ACTIVITY DIAGRAM

STATECHART DIAGRAM:

- Statechart diagram is used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.
- Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.
- Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of
- Statechart diagram is to model lifetime of an object from creation to termination.

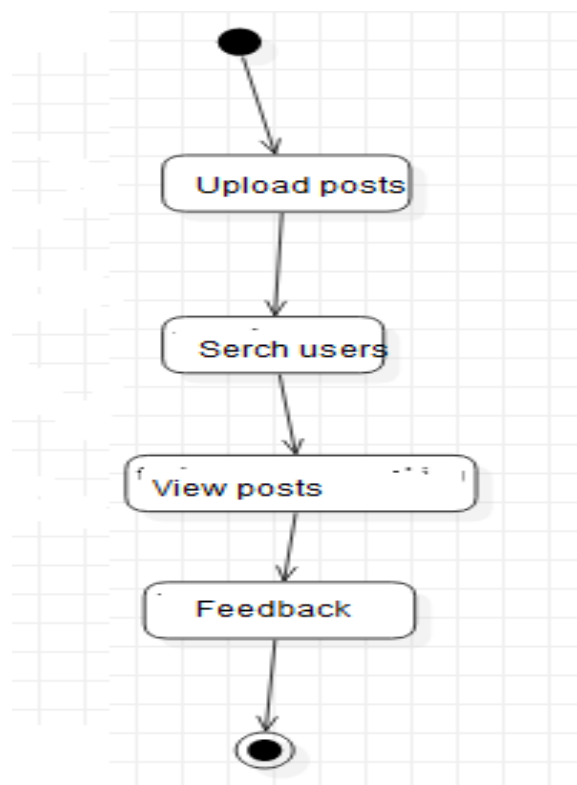


FIG:STATECHART DIAGRAM

CHAPTER - 5

TECHNOLOGY

The chapter gives the details about the technologies used.

4.1 TECHNOLOGIES USED:

PYTHON:

- Python is a popular platform used for research and development of production systems. It is a vast language with number of modules, packages and libraries that provides multiple ways of achieving a task.
- Python and its libraries like NumPy, SciPy, Scikit-Learn, Matplotlib are used in data science and data analysis. They are also extensively used for creating scalable machine learning algorithms. Python implements popular machine learning techniques such as Classification, Regression, Recommendation, and Clustering

Libraries and Packages:

To understand machine learning, you need to have basic knowledge of Python programming. In addition, there are a number of libraries and packages generally used in performing various machine learning tasks as listed below:

- numpy - is used for its N-dimensional array objects
- pandas – is a data analysis library that includes dataframes
- matplotlib – is 2D plotting library for creating graphs and plots
- scikit-learn - the algorithms used for data analysis and data mining tasks
- seaborn – a data visualization library based on matplotlib

Installation:

You can install software for machine learning in any of the two methods as discussed here:

Method 1

Download and install Python separately from python.org on various operating systems as explained below:

To install Python after downloading, double click the .exe (for Windows) or .pkg (for Mac) file and follow the instructions on the screen.

For Linux OS, check if Python is already installed by using the following command at the prompt:

\$ python --version. ...

If Python 2.7 or later is not installed, install Python with the distribution's package manager. Note that the command and package name varies.

On Debian derivatives such as Ubuntu, you can use apt:

\$ sudo apt-get install python3

Now, open the command prompt and run the following command to verify that Python is installed

\$ python3 --version

Python 3.6.2

Similarly, we can download and install necessary libraries like numpy, matplotlib etc. individually using installers like pip. For this purpose, you can use the commands shown here:

\$pip install numpy

\$pip install matplotlib

\$pip install pandas

\$pip install seaborn

Method 2

Alternatively, to install Python and other scientific computing and machine learning packages simultaneously, we should install Anaconda distribution. It is a Python implementation for Linux, Windows and OSX, and comprises various machine learning packages like numpy, scikit-learn, and matplotlib. It also includes Jupyter Notebook, an interactive Python environment. We can install Python 2.7 or any 3.x version as per our requirement.

To work with graphs and plots, we will need these Python library packages: matplotlib and seaborn.

If you are using Anaconda Python, your system already has numpy, matplotlib, pandas, seaborn, etc. installed. We start the Anaconda Navigator to access either Jupyter Note book or Spyder IDE of python.

After opening either of them, type the following commands:

```
import numpy
```

```
import matplotlib
```

Now, we need to check if installation is successful. For this, go to the command line and type in the following command:

```
$ python
```

Next, you can import the required libraries and print their versions as shown:

```
>>>import numpy
```

```
>>>print numpy.__version__
```

```
1.14.2
```

```
>>> import matplotlib
```

```
>>> print (matplotlib.__version__)
```

```
2.1.2
```

```
>> import pandas
```

```
>>> print (pandas.__version__)
```

```
0.22.0
```

```
>>> import seaborn
```

```
>>> print (seaborn.__version__)
```

```
0.8.1
```

MACHINE LEARNING:

- Machine learning is a discipline that deals with programming the systems so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs. To solve this problem, algorithms are developed that build knowledge from a specific data and past experience by applying the principles of statistical science, probability, logic, mathematical optimization, reinforcement learning, and control theory.

- Applications of Machine Learning Algorithms The developed machine learning algorithms are used in various applications such as:
 - Vision processing
 - Language processing
 - Forecasting things like stock market trends, weather
 - Pattern recognition
 - Games
 - Data mining
 - Expert systems
 - Robotics

Steps Involved in Machine Learning A machine learning project involves the following steps:

- Defining a Problem
 - Preparing Data
 - Evaluating Algorithms
 - Improving Results
 - Presenting Results
- The best way to get started using Python for machine learning is to work through a project end-to-end and cover the key steps like loading data, summarizing data, evaluating algorithms and making some predictions. This gives you a replicable method that can be used dataset after dataset. You can also add further data and improve the results.

Broadly, there are 3 types of Machine Learning Algorithms..

1.Supervised Learning

This algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, Decision Tree, Random Forest, KNN, Logistic Regression etc.

2.Unsupervised Learning

In this algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Apriori algorithm, K-means.

3. Reinforcement Learning:

Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process.

ALGORITHMS USED:

SUPPORT VECTOR MACHINE ALGORITHM:

SVM is a supervised machine learning algorithm which is mainly used to classify data into different classes. Unlike most algorithms, SVM makes use of a hyperplane which acts like a decision boundary between the various classes.

After installing Python then we have to do the following:

- Then install 'tesseract-ocr-setup-3.02.02' (this software u can find inside code folder and its used to extract text from images) software in C directory
- and then set path for tesseract-ocr-setup-3.02.02 file using below environment variable under system tab
- Variable name : tesseract
- Variable value = C:\Tesseract-OCR\tesseract.exe
- install audio to text software using below command
- pip install SpeechRecognition

TESSERACT OCR:An optical character recognition (OCR) engine

Tesseract is an OCR engine with support for unicode and the ability to recognize **more than 100 languages** out of the box. It can be trained to recognize other languages.

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License. Originally developed by Hewlett-Packard as proprietary software in the 1980s, it was released as open source in 2005 and development has been sponsored by Google since 2006.

We use nltk for obtaining whether the result is negative or positive.

What is Natural Language Processing (NLP)?

- **Natural Language Processing (NLP)** is a process of manipulating or understanding the text or speech by any software or machine. An analogy is that humans interact and understand each other's views and respond with the appropriate answer. In NLP, this interaction, understanding, and response are made by a computer instead of a human.

What is NLTK?

- **NLTK (Natural Language Toolkit)** is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.
- We will be using Python library NLTK (Natural Language Toolkit) for doing text analysis in English Language. The Natural language toolkit (NLTK) is a collection of Python libraries designed especially for identifying and tag parts of speech found in the text of natural language like English.
- Installing NLTK
- Before starting to use NLTK, we need to install it. With the help of following command, we can install it in our Python environment –
- `pip install nltk`.

For preprocessing and datamining tasks we use sklearn.

For displaying graph we use matplotlib.

Pandas is used to read the dataset.

In our project we use Django web framework.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django makes it easier to build better Web apps more quickly and with less code.

- Ridiculously fast.
 - Django was designed to help developers take applications from concept to completion as quickly as possible.
- Reassuringly secure.
 - Django takes security seriously and helps developers avoid many common security mistakes.
- Exceedingly scalable.
 - Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

We use the WAMPSERVER which is inbuilt in Django

WAMP Server:

- Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often used for web development and internal testing, but may also be used to serve live websites.
- The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet.
- WAMP also includes MySQL and PHP, which are two of the most common technologies used for creating dynamic websites.
- MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database.

- By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server.
- While Apache, MySQL, and PHP are open source components that can be installed individually, they are usually installed together.
- One popular package is called "WampServer," which provides a user-friendly way to install and configure the "AMP" components on Windows.
- **NOTE:** The "P" in WAMP can also stand for either Perl or Python, which are other scripting languages. The Mac version of LAMP is known as MAMP.
- **WE USE PYTHON FOR CODING AND AT THE BACKEND WE USE MYSQL DATABASE.**
- **THE DJANGO APPLICATION FOLLOWS MVT ARCHITECTURE.**
- The MVT (Model View Template) is a software design pattern. It is a collection of three important components Model View and Template.
- The Model helps to handle database. It is a data access layer which handles the data.
- The Template is a presentation layer which handles User Interface part completely. The View is used to execute the business logic and interact with a model to carry data and renders a template.
- Although Django follows MVC pattern but maintains it's own conventions. So, control is handled by the framework itself.
- There is no separate controller and complete application is based on Model View and Template. That's why it is called MVT application.

ABOUT MYSQL DATABASE IN PYTHON:

There are the following steps to connect a python application to our database.

1. Import mysql.connector module
2. Create the connection object.
3. Create the cursor object
4. Execute the query

Creating the connection

To create a connection between the MySQL database and the python application, the connect() method of mysql.connector module is used.

Pass the database details like HostName, username, and the database password in the method call. The method returns the connection object.

```
create table users(username varchar(50),  
password varchar(50),  
contact_no varchar(12),  
email varchar(50),  
address varchar(50));
```

```
create table postdata(username varchar(50),  
post_data varchar(500),  
post_time timestamp,  
depression varchar(50),  
motivate_post varchar(500));
```

Databases SQLite

Name
Online depression

SQLite Database

Tables

Name	Description
Users	Contains all the registered user details.
Upload posts	All the registered service provider details.
Services	Contains all the types of services available.

List of Database Tables

CHAPTER - 6

CODING

This chapter gives an overview of al the project implementation.

6.1 IMPLEMENTATION:

```
import pymysql
print(pymysql.__file__)
pymysql.install_as_MySQLdb()
```

''''''

Django settings for Depression project.

Generated by 'django-admin startproject' using Django 2.2.7.

*For more information on this file, see
<https://docs.djangoproject.com/en/2.2/topics/settings/>*

*For the full list of settings and their values, see
<https://docs.djangoproject.com/en/2.2/ref/settings/>*

''''''

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '%8f!97e_yfkjc*t1s2@p6f1z531f*fm0t@2dmo9p74q#!tza%ou'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'DepressionApp'
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
```

```
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'Depression.urls'
```

```
TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [
os.path.join('C:/Python/Depression/DepressionApp', 'templates'),
],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
],
},
},
]
```

```
WSGI_APPLICATION = 'Depression.wsgi.application'
```

```
CACHES = {
'default': {
'BACKEND': 'django.core.cache.backends.filebased.FileBasedCache',
'LOCATION': 'C:/cache/django_cache.txt',
}
}
```

```
# Database
```

```
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases
```

```
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.mysql',
'NAME': 'depression',
'HOST': '127.0.0.1',
'PORT': '3308',
'USER': 'root',
'PASSWORD': 'root',
'OPTIONS': {
'autocommit': True,
},
}
}
```

```
# Password validation
```

```
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators
```

```
AUTH_PASSWORD_VALIDATORS = [
```



```
{
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
}
```

```
# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/
```

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/
```

```
STATIC_URL = '/static/'
```

""""Depression URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see: <https://docs.djangoproject.com/en/2.2/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `path('', Home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

```
""""
```

```
from django.contrib import admin
from django.urls import path, include
```

```
urlpatterns = [
```

```
path('admin/', admin.site.urls),
path("", include('DepressionApp.urls')),
]
```

WSGI config for Depression project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

<https://docs.djangoproject.com/en/2.2/howto/deployment/wsgi/>

```
import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Depression.settings')

application = get_wsgi_application()
```

/*

Design by Free CSS Templates

<http://www.freecsstemplates.org>

Released for free under a Creative Commons Attribution 3.0 License

Name : Big Business

Description: A two-column, fixed-width design with a bright color scheme.

Version : 1.0

Released : 20120210

*/

```
* {
    margin: 0;
    padding: 0;
}

a {
    text-decoration: underline;
    color: #0F8C8C;
}

a:hover {
    text-decoration: none;
}
```

```
body {
    font-size: 11.5pt;
    color: #5C5B5B;
    line-height: 1.75em;
    background: #E0DCDC
    url(images/img01.gif) repeat-x top left;
}
```

```

body,input {
    font-family: Georgia, serif;
}

strong {
    color: #2C2B2B;
}

br.clearfix {
    clear: both;
}

h1,h2,h3,h4 {
    font-weight: normal;
    letter-spacing: -1px;
}

h2 {
    font-size: 2.25em;
}

h2,h3,h4 {
    color: #2C2B2B;
    margin-bottom: 1em;
}

h3 {
    font-size: 1.75em;
}

h4 {
    font-size: 1.5em;
}

img.alignleft {
    margin: 5px 20px 20px 0;
    float: left;
}

img.aligntop {
    margin: 5px 0 20px 0;
}

img.pic {
    padding: 5px;
    border: solid 1px #D4D4D4;
}

p {
    margin-bottom: 1.5em;
}

ul {
    margin-bottom: 1.5em;
}

```

```

ul h4 {
    margin-bottom: 0.35em;
}

.box {
    overflow: hidden;
    margin-bottom: 1em;
}

.date {
    background: #6E6E6E;
    padding: 5px 6px 5px 6px;
    margin: 0 6px 0 0;
    color: #FFFFFF;
    font-size: 0.8em;
    border-radius: 2px;
}

#content {
    width: 665px;
    float: left;
    padding: 0;
}

#content-box1 {
    width: 320px;
    float: left;
}

#content-box2 {
    margin: 0 0 0 345px;
    width: 320px;
}

#footer {
    margin: 40px 0 120px 0;
    text-align: center;
    color: #8C8B8B;
}

#footer a {
    color: #8C8B8B;
}

#header {
    height: 75px;
    position: relative;
    background: #6E6E6E url(images/img03.jpg) top left no-repeat;
    padding: 45px;
    color: #FFFFFF;
    width: 888px;
    border: solid 1px #7E7E7E;
    border-top-left-radius: 5px;
    border-top-right-radius: 5px;
    overflow: hidden;
}

```

```

#logo {
line-height: 160px;
height: 160px;
padding: 5px 0 0 0;
position: absolute;
top: 0;
left: 45px;
}

#logo a {
text-decoration: none;
color: #FFFFFF;
text-shadow: 0 1px 1px #3E3E3E;
}

#logo h1 {
font-size: 1.75em;
}

#slogan {
line-height: 160px;
height: 160px;
padding: 5px 0 0 0;
position: absolute;
right: 45px;
top: 0;
}

#slogan h2 {
color: #BEBEBE;
font-size: 1.25em;
text-shadow: 0 1px 1px #3E3E3E;
}

#menu {
padding: 0 45px 0 45px;
position: relative;
background: #209D9D url(images/img02.gif) repeat-x top left;
margin: 0 0 0 0;
height: 60px;
line-height: 60px;
width: 890px;
border-top: solid 1px #5AD7D7;
text-shadow: 0 1px 1px #007D7D;
}

#menu a {
text-decoration: none;
color: #FFFFFF;
font-size: 1.25em;
letter-spacing: -1px;
}

#menu ul {
list-style: none;
}

```

#menu ul li {	
	padding: 0 20px 0 20px;
}	display: inline;
#menu ul li.first {	
}	padding-left: 0;
#page {	
	padding: 45px 45px 15px 45px;
	position: relative;
	width: 890px;
	margin: 0;
}	
#page .section-list {	
	list-style: none;
}	padding-left: 0;
#page .section-list li {	
	clear: both;
}	padding: 30px 0 30px 0;
#page ul {	
}	list-style: none;
#page ul li {	
	border-top: solid 1px #D4D4D4;
}	padding: 15px 0 15px 0;
#page ul li.first {	
	padding-top: 0;
}	border-top: 0;
#page-bottom {	
	position: relative;
	margin: 0;
	background: #6E6E6E
url(images/img03.jpg) top left no-repeat;	
	border: solid 1px #7E7E7E;
	width: 888px;
	padding: 45px 45px 0 45px;
	color: #DCDCDC;
	border-bottom-left-radius: 5px;
	border-bottom-right-radius: 5px;
}	
#page-bottom a {	
	color: #F5F5F5;
}	

```

#page-bottom h2, #page-bottom h3, #page-bottom h4 {
    color: #FFFFFFF;
}

#page-bottom ul {
    list-style: none;
}

#page-bottom ul li {
    border-top: solid 1px #8F8F8F;
    padding: 15px 0 15px 0;
}

#page-bottom ul li.first {
    padding-top: 0;
    border-top: 0;
}

#page-bottom-content {
    width: 665px;
    float: left;
}

#page-bottom-sidebar {
    width: 200px;
    margin: 0 0 0 690px;
}

#search input.form-submit {
    margin-left: 1em;
    color: #FFFFFFF;
    padding: 10px;
    background: #2FACAC;
    border: 0;
}

#search input.form-text {
    border: solid 1px #8F8F8F;
    padding: 10px;
}

#sidebar {
    width: 200px;
    padding: 0;
    margin: 0 0 0 690px;
}

#splash {
    margin: 0 0 0 0;
    height: 250px;
    position: relative;
    padding: 45px 45px 10px 45px;
    width: 890px;
}

#splash .pic {

```

```

padding: 9px;
}

#wrapper {
position: relative;
width: 980px;
margin: 75px auto 0 auto;
background: #FFFFFF;
}

/*
CSS Credit: http://www.templatemo.com
*/

body {
margin: 0;
padding: 0;
line-height: 1.5em;
font-family: Arial, Helvetica, sans-serif;
font-size: 15px;
color: black;
background: #afa87d;
}

a:link, a:visited { color: #d46528; text-decoration: none; font-weight: bold; }
a:active, a:hover { color: #2da3e9; }

p {
margin: 0px;
padding: 0px;
}

img {
margin: 0px;
padding: 0px;
border: none;
}

.cleaner { clear: both; width: 100%; height: 0px; font-size: 0px; }

.margin_bottom_10 { float: left; width: 100%; height: 10px; font-size: 1px; }
.margin_bottom_20 { clear: both; width: 120%; height: 25px; font-size: 19px; }
.margin_bottom_30 { clear: both; width: 100%; height: 30px; font-size: 1px; }
.margin_bottom_40 { clear: both; width: 100%; height: 40px; font-size: 1px; }
.margin_bottom_50 { clear: both; width: 100%; height: 50px; font-size: 1px; }
.margin_bottom_60 { clear: both; width: 100%; height: 60px; font-size: 1px; }

.margin_right_10 { margin-right: 10px; }
.margin_right_20 { margin-right: 20px; }
.margin_right_50 { margin-right: 50px; }

.border_bottom {
border-bottom: 1px solid #CCC;
}

```



```

#templatemo_container {
width: 1040px;
margin: 0 auto;
}

#templatemo_header {
width: 1040px;
height: 250px;
background:
url(images/templatemo_header_bg.jpg) no-repeat;
}

#templatemo_header #site_logo {
float: left;
margin: 80px 0 0 40px;
width: 410px;
height: 75px;
background:url(images/logo1.png)
bottom no-repeat;
}

.rc_btn_01 a{
float: right;
clear: both;
display: block;
width: 80px;
height: 15px;
text-align: center;
padding: 10px 0 10px 10px;
background:
url(images/templatemo_bottom_01.jpg) bottom right no-repeat;
color: #d46528;
font-weight: bold;
text-decoration: none;
}
/* menu */

#templatemo_menu {
clear: both;
width: 920px;
height: 50px;
padding: 0 80px 0 40px;
background:
url(images/templatemo_menu_bg.jpg) no-repeat;
}

#templatemo_menu ul {
margin: 0px;
padding: 0px;
list-style: none;
}

#templatemo_menu ul li {
display: inline;
}

```

#templatemo_menu ul li a {	float: left;
	padding: 20px 40px 0 0;
	text-align: center;
	font-size: 12px;
	text-align: center;
	text-decoration: none;
	color: #2aa3e8;
	font-weight: bold;
	outline: none;
}	
#templatemo_menu li a:hover, #templatemo_menu li .current {	
}	color: #000000;
#templatemo_menu li .last {	
}	background: none;
/* end of menu*/	
/* content */	
.header_01 {	
	font-size: 20px;
	padding-bottom: 10px;
	margin-bottom: 20px;
	font-weight: bold;
	color: #d46528;
}	
#templatemo_content {	
	clear: both;
	width: 920px;
	padding: 10px 80px 30px 40px;
	background:
url(images/templatemo_content_bg.jpg) repeat-y;	
}	
#templatemo_content #content_left {	
	float: left;
	width: 600px;
	padding: 20px 0 0 0;
	/* background: #a4ddfe; */
}	
#templatemo_content #content_right {	
	float: right;
	width: 270px;
}	
#content_left .left_column_section {	
	margin: 0;
	padding: 0;
}	

<pre>.left_column_section p {</pre>	<pre>text-align: justify; margin-bottom: 10px;</pre>
<pre>}</pre>	
<pre>.image_box {</pre>	<pre>float: left; width: 280px; height: 120px; background: #ffffff; border: 1px solid #999; padding: 4px;</pre>
<pre>}</pre>	
<pre>.image_box img {</pre>	<pre>width: 280px; height: 120px;</pre>
<pre>}</pre>	
<pre>#content_right .right_column_section {</pre>	<pre>clear: both; position: relative; background:</pre>
<pre>url(images/templatemo_section_01_content_bg.jpg) repeat-y; }</pre>	
<pre>.right_column_section .header_02 {</pre>	<pre>width: 270px; height: 30px; font-size: 16px; font-weight: bold; padding: 40px 0 0 0; text-align: center; background:</pre>
<pre>url(images/templatemo_section_01_header_bg.jpg) no-repeat; }</pre>	
<pre>.right_column_section .header_03 {</pre>	<pre>font-size: 12px; margin-bottom: 5px; font-weight: bold; color: #333333;</pre>
<pre>}</pre>	
<pre>.right_column_section .content {</pre>	<pre>padding: 10px 20px 0 20px; background:</pre>
<pre>url(images/templatemo_section_01_content_bg.jpg) repeat-y; }</pre>	
<pre>.right_column_section span {</pre>	<pre>position: absolute; width: 270px; height: 60px;</pre>

```

url(images/templatemo_section_01_bottom_bg.jpg);
}

/* bottom panel */

#templatemo_bottom_panel {

url(images/templatemo_bottom_panel_bg.jpg) no-repeat;
}

.content_panel_section {

}

.content_panel_section ul {

}

.content_panel_4_col li {

}

.content_panel_4_col li a {

}

.content_panel_4_col li a:hover {

}

.content_panel_4_col li span {

}

/* end of bottom panel*/

/* footer */
#templatemo_footer {

```

background:

clear: both;
width: 600px;
height: 210px;
padding: 0px 400px 30px 40px;
background:

float: left;
width: 275px;

margin: 0px;
padding: 0px;
list-style: none;

margin: 0px;
padding: 0px;
color: #2da3e9;
padding-bottom: 5px;
margin-bottom: 5px;
border-bottom: 1px solid #CCC;

color: #2da3e9;

color: #d46528;

clear: both;
display: block;
color: #333;
font-weight: normal;

clear: both;
width: 920px;

```
padding: 20px 80px 30px 40px;
text-align: left;
color: #000;
background: #afa87d
```

```
url(images/templatemo_footer.jpg) top center no-repeat;
}
```

```
#templatemo_footer a{
```

```
color: #000;
font-weight: bold;
```

```
}
/* end of footer */
```

ADMIN.PY

```
from django.contrib import admin
```

```
# Register your models here.
```

APPS.PY

```
from django.apps import AppConfig
```

```
class DepressionappConfig(AppConfig):
    name = 'DepressionApp'
```

MODELS.PY

```
from django.db import models
```

```
# Create your models here.
```

TESTS.PY

```
from django.test import TestCase
```

```
# Create your tests here.
```

URLS

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [path("index.html", views.index, name="index"),
views.Register, name="Register"),
name="Signup"),
views.Login, name="Login"),
views.UserLogin, name="UserLogin"),
views.SearchFriends, name="SearchFriends"),
views.UploadPost, name="UploadPost"),
views.UploadPostData, name="UploadPostData"),
views.AdminLogin, name="AdminLogin"),
```

```
path("Register.html",
path("Signup", views.Signup,
path("Login.html",
path("UserLogin",
path("SearchFriends.html",
path("UploadPost.html",
path("UploadPostData",
path("AdminLogin",
```

```

views.Admin, name="Admin"),
views.ViewUsers, name="ViewUsers"),
views.ViewPosts, name="ViewPosts"),

path("SendMotivatedPost.html", views.SendMotivatedPost, name="SendMotivatedPost"),
path("SendMotivatedPostData", views.SendMotivatedPostData, name="SendMotivatedPostData"),
path("ViewMotivatedPost.html", views.ViewMotivatedPost, name="ViewMotivatedPost"),
views.MotivatedText, name="MotivatedText"),

]

```

VIEWS

```

from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
import pymysql
from django.http import HttpResponse
from django.conf import settings
from django.core.files.storage import FileSystemStorage
import datetime
from sklearn.externals import joblib
import PIL.Image
import pytesseract
import matplotlib.pyplot as plt
import re
import numpy as np
import speech_recognition as sr
# Create your views here.

svm_classifier = joblib.load('svmClassifier.pkl')

def index(request):
    if request.method == 'GET':
        return render(request, 'index.html', {})

def UploadPost(request):
    if request.method == 'GET':
        return render(request, 'UploadPost.html', {})

def Register(request):
    if request.method == 'GET':
        return render(request, 'Register.html', {})

def Admin(request):
    if request.method == 'GET':
        return render(request, 'Admin.html', {})

def Login(request):
    if request.method == 'GET':
        return render(request, 'Login.html', {})

```

```

def SendMotivatedPost(request):
    if request.method == 'GET':
        return render(request, 'SendMotivatedPost.html', {})

def predict(textdata,classifier):
    text_processed = textdata
    X = [text_processed]
    sentiment = classifier.predict(X)
    return (sentiment[0])

def predictSentiment(textdata):
    result = predict(textdata, svm_classifier)
    predicts = ""
    if result == 0:
        predicts = "Negative"
    if result == 1:
        predicts = "Positive"
    return predicts

def SendMotivatedPostData(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        time = request.POST.get('t2', False)
        text = request.POST.get('t3', False)
        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root',
        database = 'depression',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "update postdata set motivate_post='"+text+"' where
        username='"+username+"' and post_time='"+time+"' and motivate_post='Pending'"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        context= { 'data': 'Your motivated text sent to user '+username }
        return render(request, 'SendMotivatedPost.html', context)

def UploadPostData(request):
    if request.method == 'POST' and request.FILES['t1']:
        output = ""
        myfile = request.FILES['t1']
        fs = FileSystemStorage()
        name = str(myfile)
        if name.lower().endswith(('.txt')):
            name = 'text.txt'
        elif name.lower().endswith(('.png', '.jpg', '.jpeg', 'gif')):
            name = 'img.jpg'
        filename = fs.save(name, myfile)
        if name.lower().endswith(('.txt')):
            with open("text.txt", "r") as file:
                for line in file:
                    line = line.strip('\n')
                    output+=line+' '
        elif name.lower().endswith(('.png', '.jpg', '.jpeg', 'gif')):
            output = pytesseract.image_to_string(PIL.Image.open(name))
            output = output.replace('\n',' ')
        elif name.lower().endswith(('.wav')):

```

```

    r = sr.Recognizer()
    with sr.WavFile(name) as source:
        audio = r.record(source)
    try:
        output = r.recognize_google(audio)
    except:
        pass
    user = "
    with open("session.txt", "r") as file:
        for line in file:
            user = line.strip('\n')
    now = datetime.datetime.now()
    option = 'Pending'
    output = re.sub('\W+', '', output)
    current_time = now.strftime("%Y-%m-%d %H:%M:%S")
    sentiment = predictSentiment(output.lower())
    db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root',
    database = 'depression',charset='utf8')
    db_cursor = db_connection.cursor()
    student_sql_query = "INSERT INTO
    postdata(username,post_data,post_time,depression,motivate_post)
    VALUES('"+user+"','"+output+"','"+current_time+"','"+sentiment+"','"+option+"')"
    db_cursor.execute(student_sql_query)
    db_connection.commit()
    print(db_cursor.rowcount, "Record Inserted")
    if db_cursor.rowcount == 1:
        context= {'data': 'Detected Depression From Uploaded File : '+sentiment}
        return render(request, 'UploadPost.html', context)
    else:
        context= {'data': 'Error in signup process'}
        return render(request, 'UploadPost.html', context)

```

```

def ViewUsers(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center
        width=100%><tr><th>Username</th><th>Password</th><th>Contact No</th><th>Email
        ID</th><th>Address</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
        'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:
                strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+row[2]+'</td><td>'+str(row[3])+'</td><td>'
                +str(row[4])+'</td></tr>'
            context= {'data':strdata}
            return render(request, 'ViewUsers.html', context)

```

```

def ViewPosts(request):
    if request.method == 'GET':
        positive = 0
        negative = 0
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
        Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'

```



```

con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
with con:
    cur = con.cursor()
    cur.execute("select * FROM postdata")
    rows = cur.fetchall()
    for row in rows:
        if row[3] == 'Negative':
            negative = negative + 1
        else:
            positive = positive + 1

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td><
td>'+str(row[4])+'</td></tr>'
height = [positive,negative]
bars = ('Depression Posts', 'Non Depression Post')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.show()
context= {'data':strdata}
return render(request, 'ViewPosts.html', context)

def MotivatedText(request):
    if request.method == 'GET':
        user = "
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip("\n")
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM postdata")
            rows = cur.fetchall()
            for row in rows:
                if row[0] == user:

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td><
td>'+str(row[4])+'</td></tr>'
            context= {'data':strdata}
            return render(request, 'MotivatedText.html', context)

def ViewMotivatedPost(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM postdata")
            rows = cur.fetchall()
            for row in rows:
                if row[4] != 'Pending':

```

```

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td><td>'+str(row[4])+'</td></tr>'
context= {'data':strdata}
return render(request, 'ViewMotivatedPost.html', context)

def SearchFriends(request):
    if request.method == 'GET':
        user = ""
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip('\n')
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Contact
No</th><th>Email ID</th><th>Address</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:
                if row[0] != user:

strdata+='<td>'+row[0]+'</td><td>'+row[2]+'</td><td>'+row[3]+'</td><td>'+str(row[4])+'</td></tr>'
context= {'data':strdata}
return render(request, 'SearchFriends.html', context)

def UserLogin(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        index = 0
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:
                if row[0] == username and password == row[1]:
                    index = 1
                    break
        if index == 1:
            file = open('session.txt','w')
            file.write(username)
            file.close()
            context= {'data': 'welcome '+username}
            return render(request, 'UserScreen.html', context)
        else:
            context= {'data': 'login failed'}
            return render(request, 'Login.html', context)

def Signup(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)

```

```

contact = request.POST.get('t3', False)
email = request.POST.get('t4', False)
address = request.POST.get('t5', False)
db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root',
database = 'depression',charset='utf8')
db_cursor = db_connection.cursor()
student_sql_query = "INSERT INTO users(username,password,contact_no,email,address)
VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"
db_cursor.execute(student_sql_query)
db_connection.commit()
print(db_cursor.rowcount, "Record Inserted")
if db_cursor.rowcount == 1:
    context= {'data':'Signup Process Completed'}
    return render(request, 'Register.html', context)
else:
    context= {'data':'Error in signup process'}
    return render(request, 'Register.html', context)

```

```

def AdminLogin(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        if username == 'admin' and password == 'admin':
            context= {'data':'welcome '+username}
            return render(request, 'AdminScreen.html', context)
        else:
            context= {'data':'login failed'}
            return render(request, 'Admin.html', context)

```

MANAGE.PY (THIS IS THE MAIN CODE THAT IS EXECUTED IN COMMAND PROMPT)

```

#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Depression.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

TO RUN THE CODE WE USE THE FOLLOWING COMMAND:

```
python manage.py runserver
```

CHAPTER - 7

TESTING

The chapter shows the various test cases.

6.1 ABOUT TESTING:

- The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of testings are:

- Unit testing
 - Integration testing
 - Acceptance testing
 - Smoke testing
 - Interface testing
-
- **Software Testing:**
 - Software testing is the process of validating and verifying that a software application meets the technical requirements which are involved in its design and development. It is also used to uncover any defects/bugs that exist in the application. It assures the quality of the software. There are many types of testing software viz., manual testing, unit testing, black box testing, performance testing, stress testing, regression testing, white box testing etc. Among these performance testing and load testing are the most important one for an android application and next sections deal with some of these types.

- ***Black box Testing***

- Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing, and specification-based testing.

- ***White box Testing***

- White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

- ***Performance Testing***

- Performance testing is executed to determine how fast a system or sub-system performs under a particular workload. It can also serve to validate and verify other quality attributes of the system such as scalability, reliability and resource usage.

- ***Load Testing***

- Load testing is primarily concerned with testing that can continue to operate under specific load, whether that is large quantities of data or a large number of users.

- ***Manual Testing***

- Manual Testing is the process of manually testing software for defects. Functionality of this application is manually tested to ensure the correctness. Few examples of test case for Manual Testing are discussed later in this chapter.

- **Example: Testing** the end-to-end flow of a piece of software. For **example**, a real estate application that allows users to upload photos and create real estate listings – acceptance **testing** should verify this can be done.

6.2TEST CASES:

Test Case 1	
Test Case Name	Empty login fields testing
Description	In the login screen if the username and password fields are empty
Output	Login fails showing an alert box asking to enter username and password.

Table 6:1 Test Case for Empty Login Fields

Test Case 2	
Test Case Name	Wrong login fields testing
Description	A unique username and password are set by administrator. On entering wrong username or password gives.
Output	Login fails showing an alert box username or password incorrect.

Table 6:2 Test Case for Wrong Login Fields

Test Case 3	
Test Case Name	User Signup Fails.
Description	User signup need to provide all data.
Output	Signup Fails and an alert message appears asking to enter valid email and name.

Table 6:3 Test Case for Signup fail

CHAPTER - 8

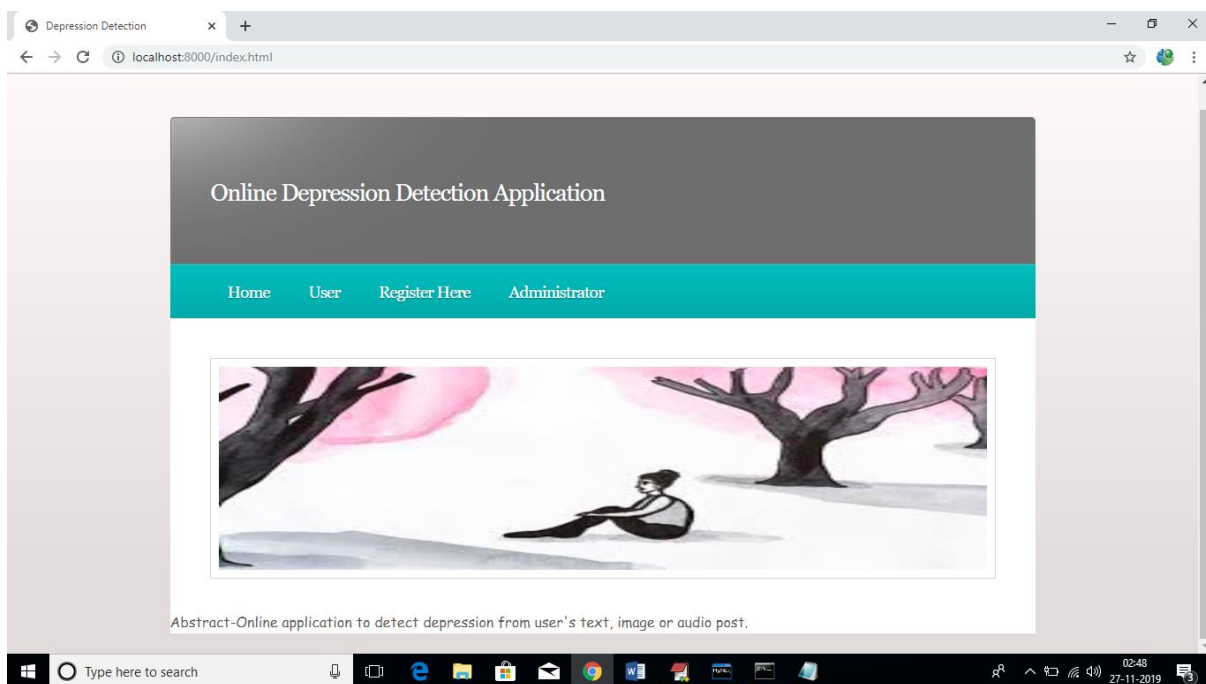
OUTPUT SCREENS

This Chapter shows us various output screens

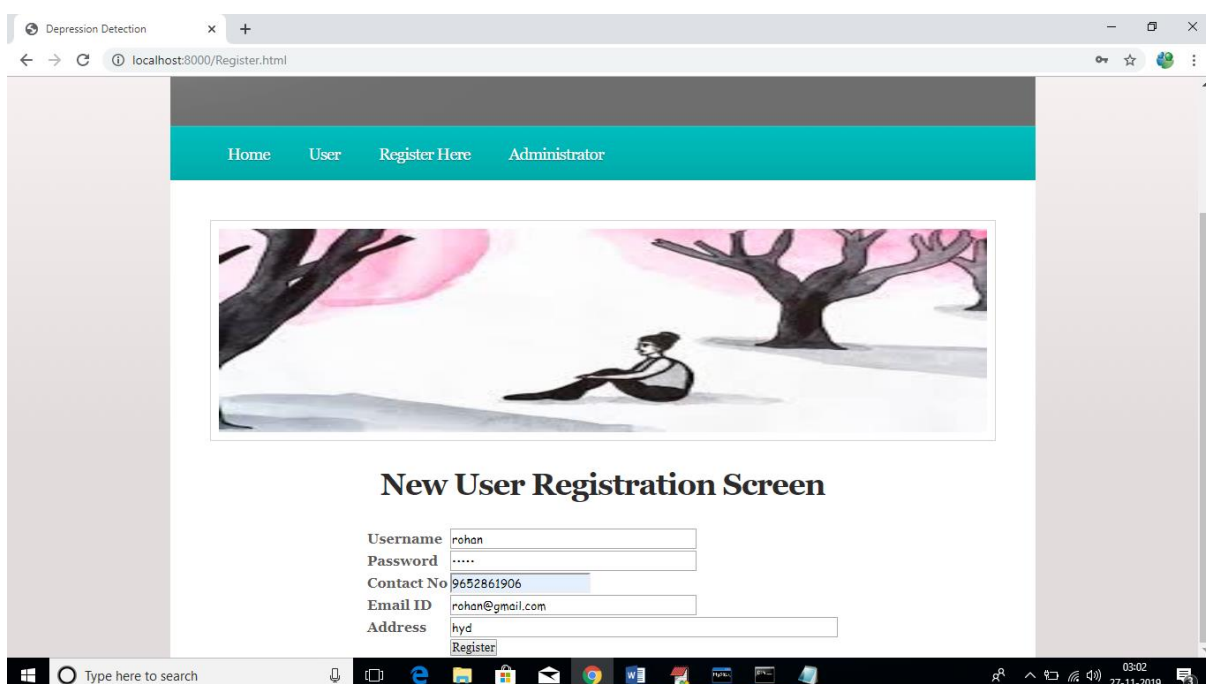
8.1 Screen Captures

8.1.1 User Login Screen:

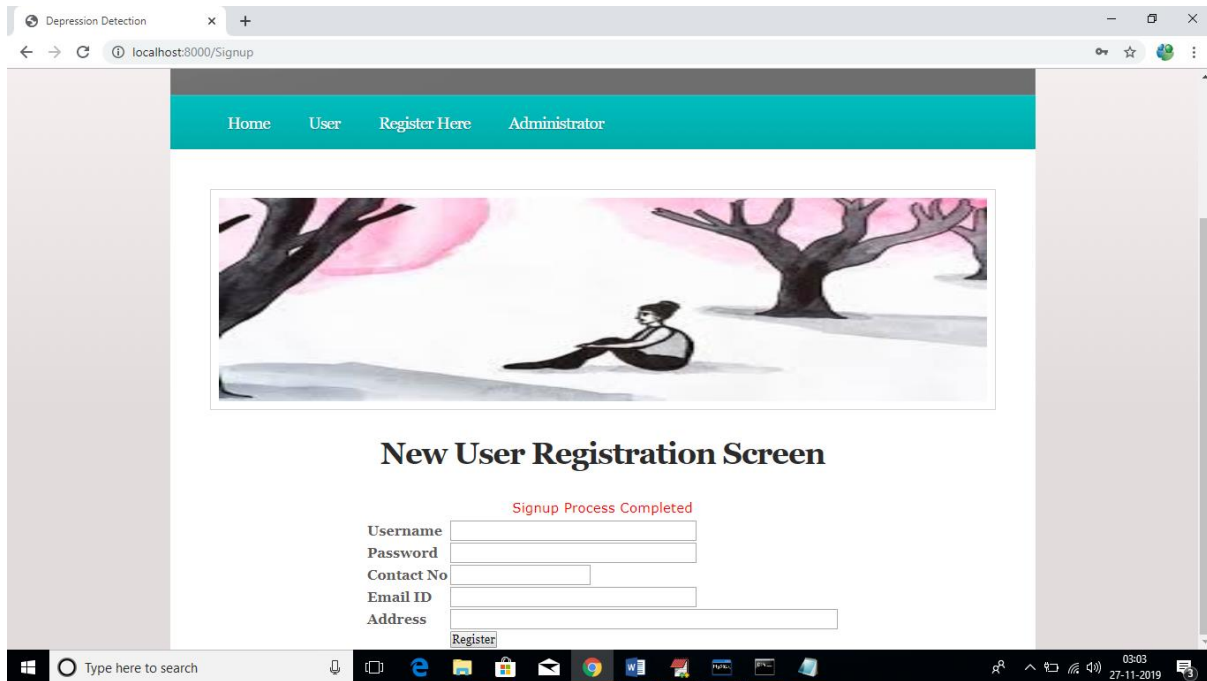
Now deploy code in DJANGO and start server and run in browser to get below screen



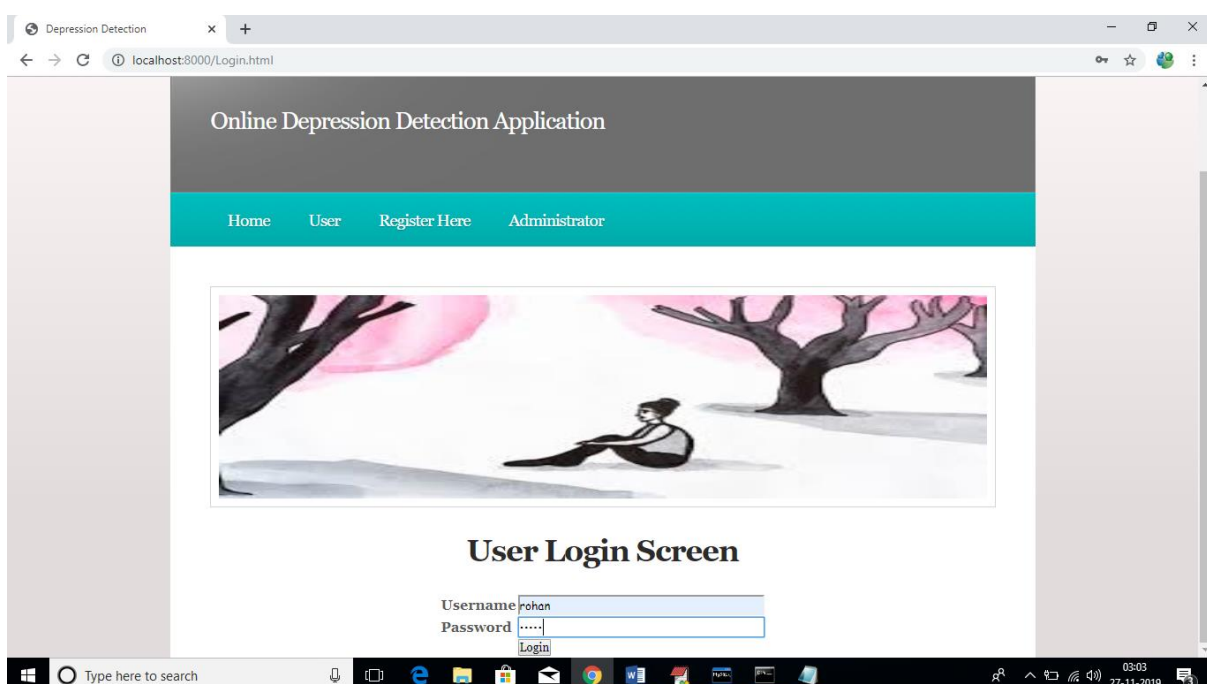
In above screen click on 'Register Here' link to add new user to application



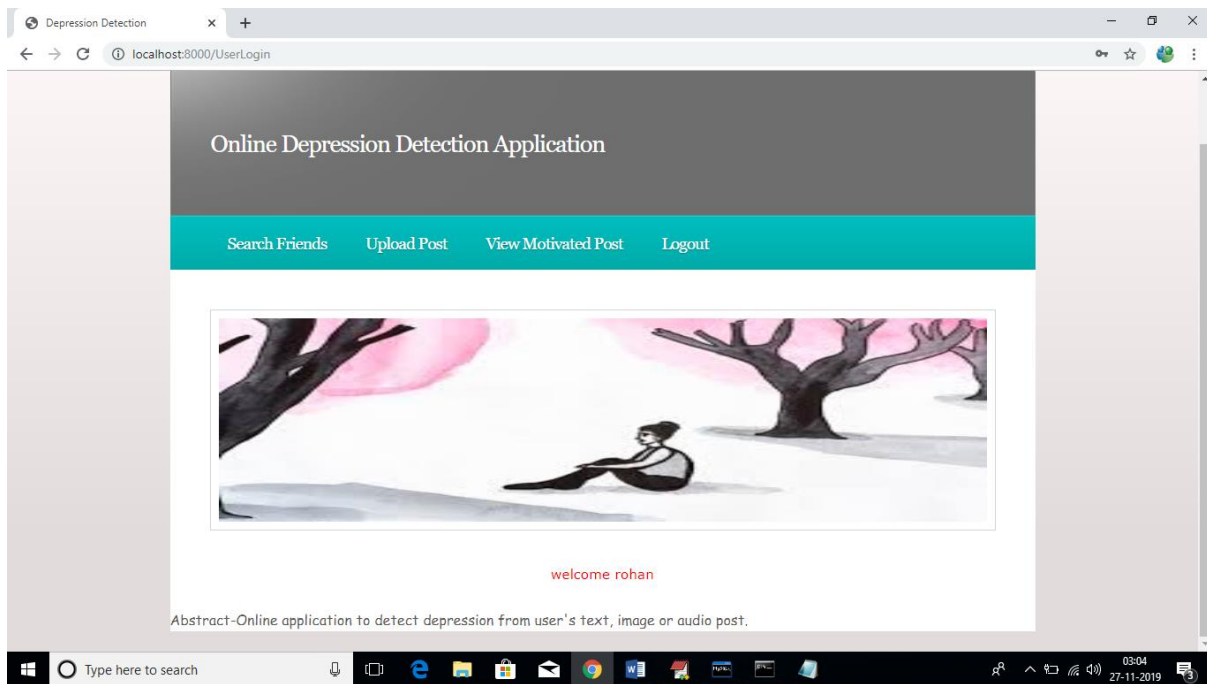
After register will get below confirmation screen



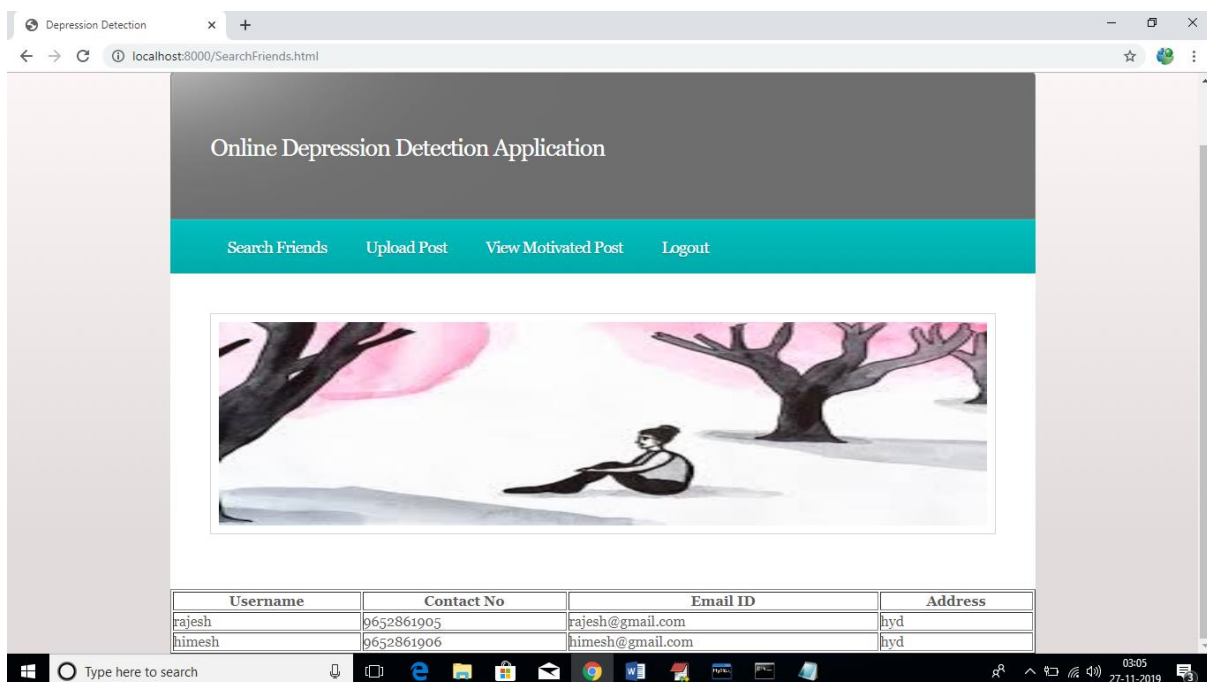
Now click on 'User' link to login as 'user'



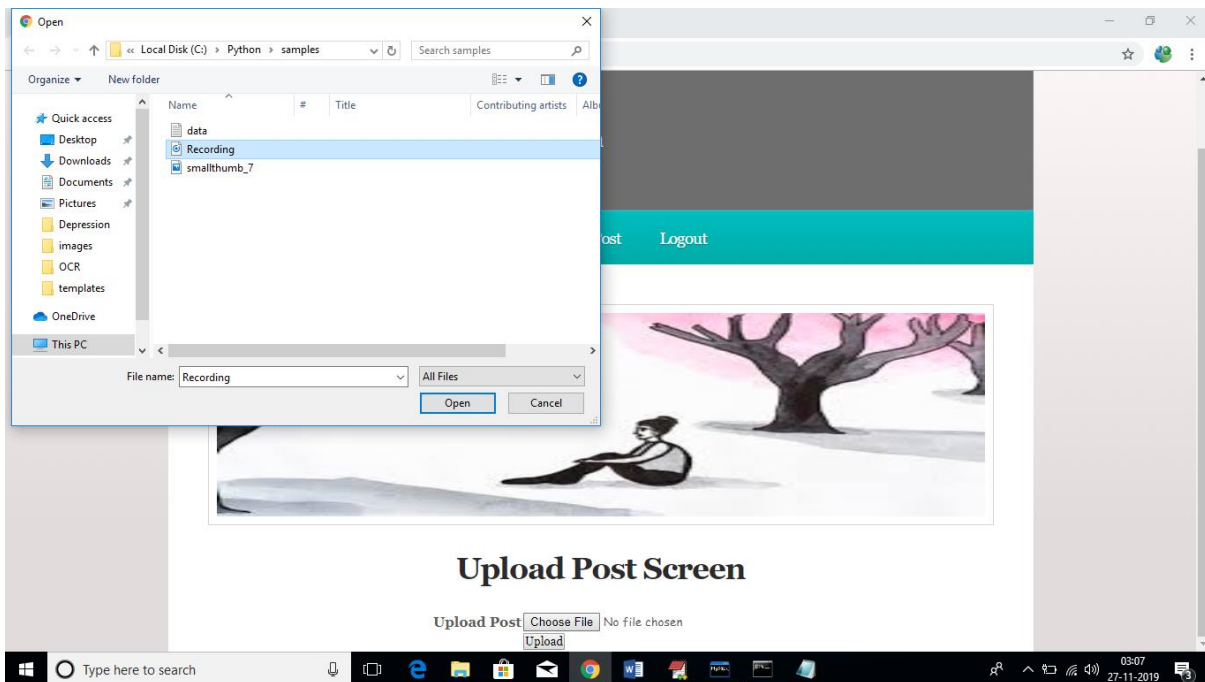
After login will get below user screen



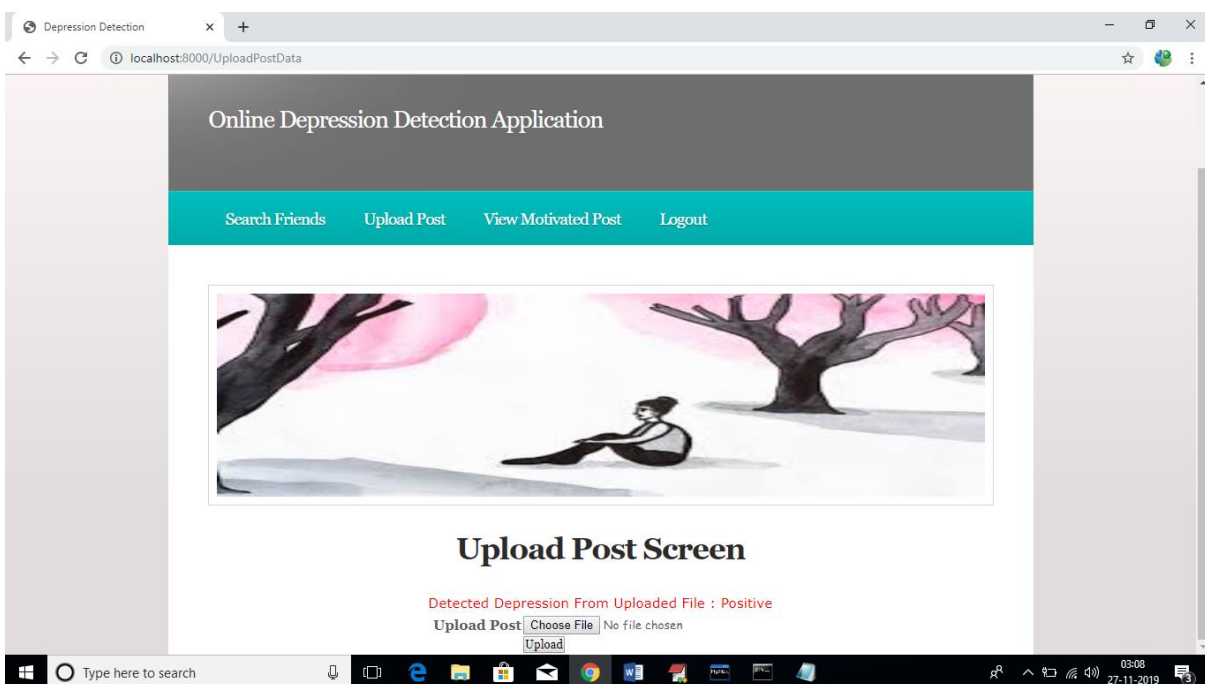
In above screen click on 'Search Friends' link to view all users registered with the application



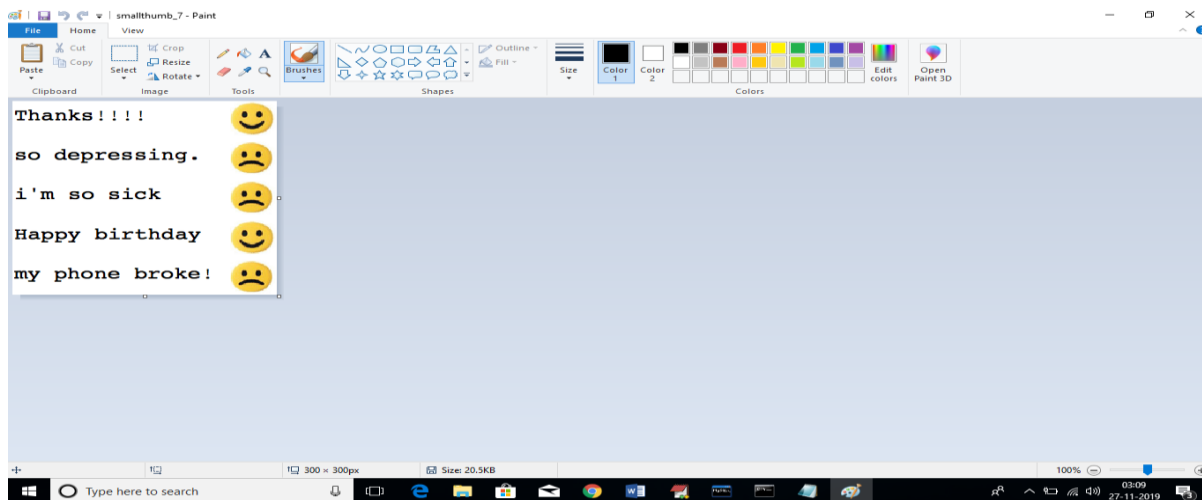
Now click on 'Upload Post' link to upload post files such as text, image or audio



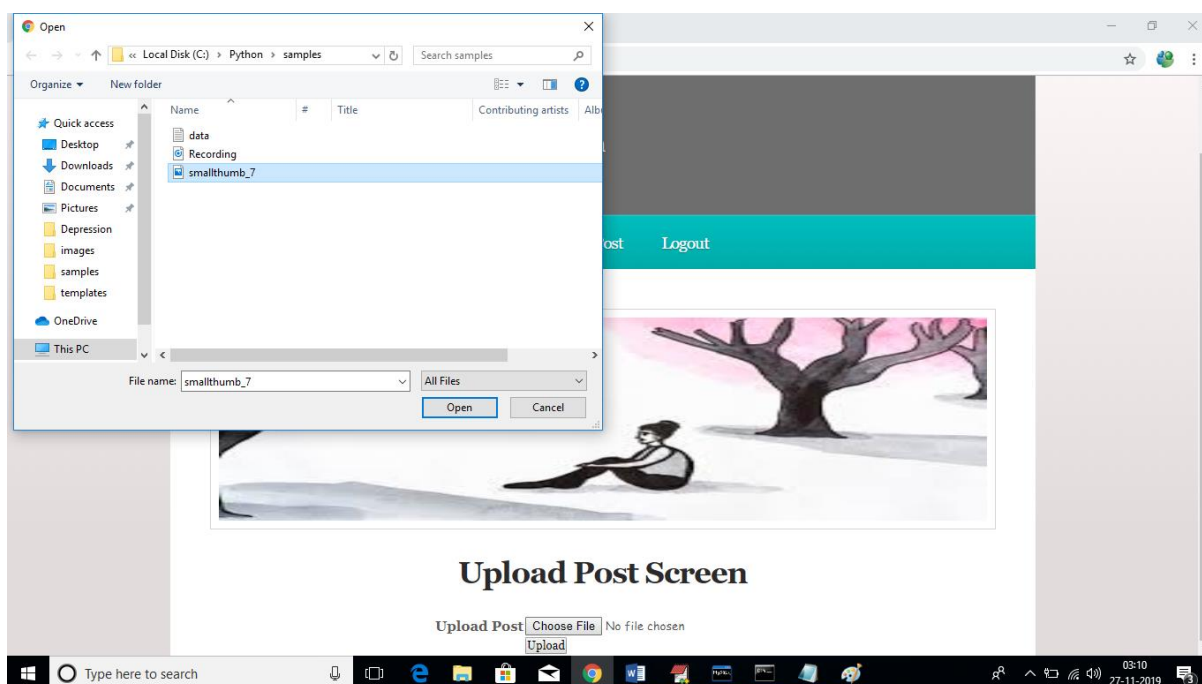
In above screen I am uploading one recording file, after upload will get below message from recording data



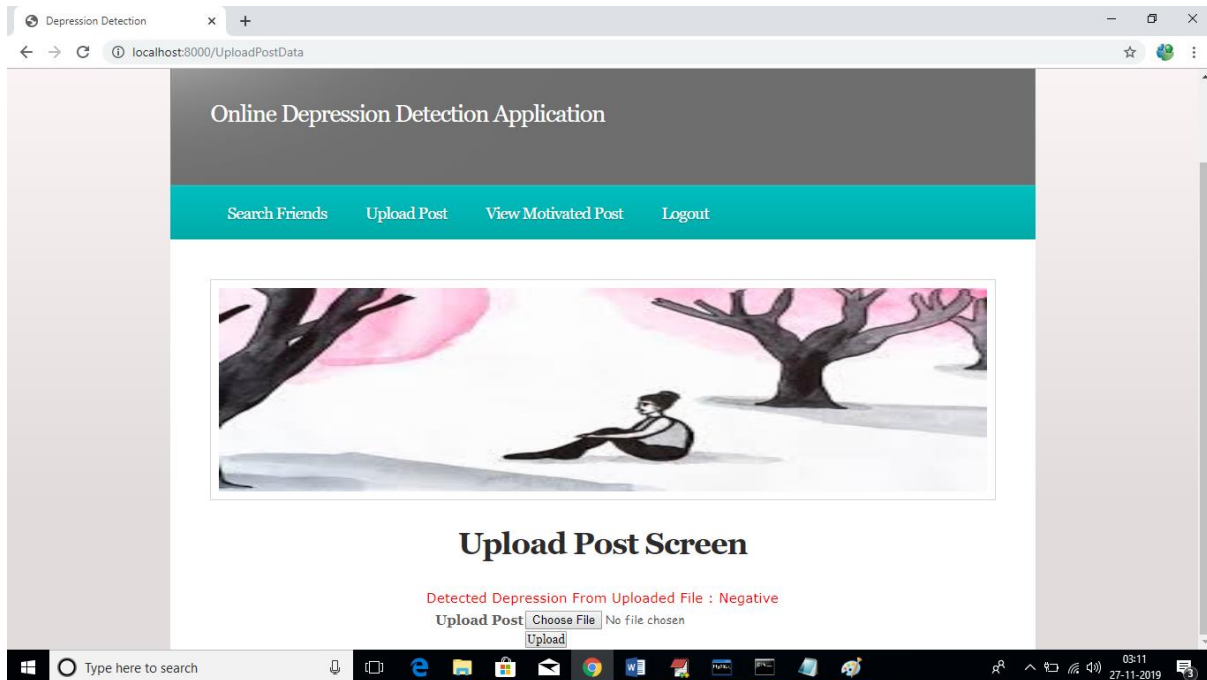
In above screen its says uploaded file contains message which indicate user is happy and gave positive recording. In below screen I am uploading one image which contains depress negative message



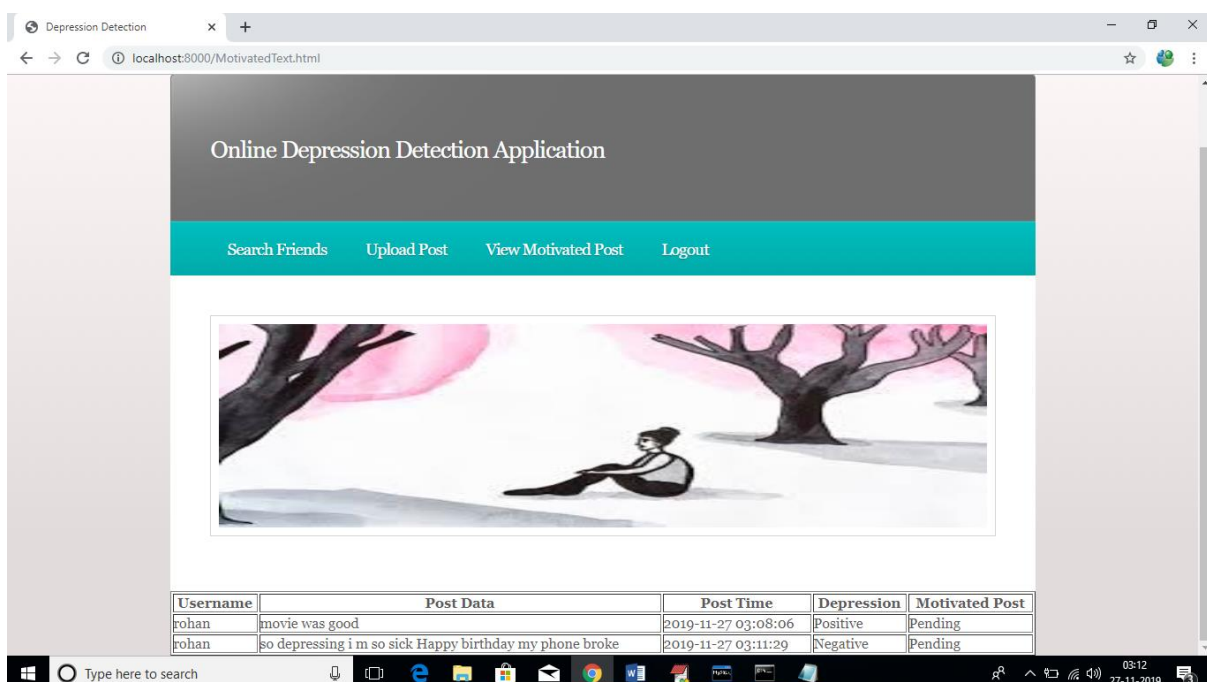
In above screen in title image file name is 'smallthumb-7' and it contains depress negative message and I will upload same image to application and see results



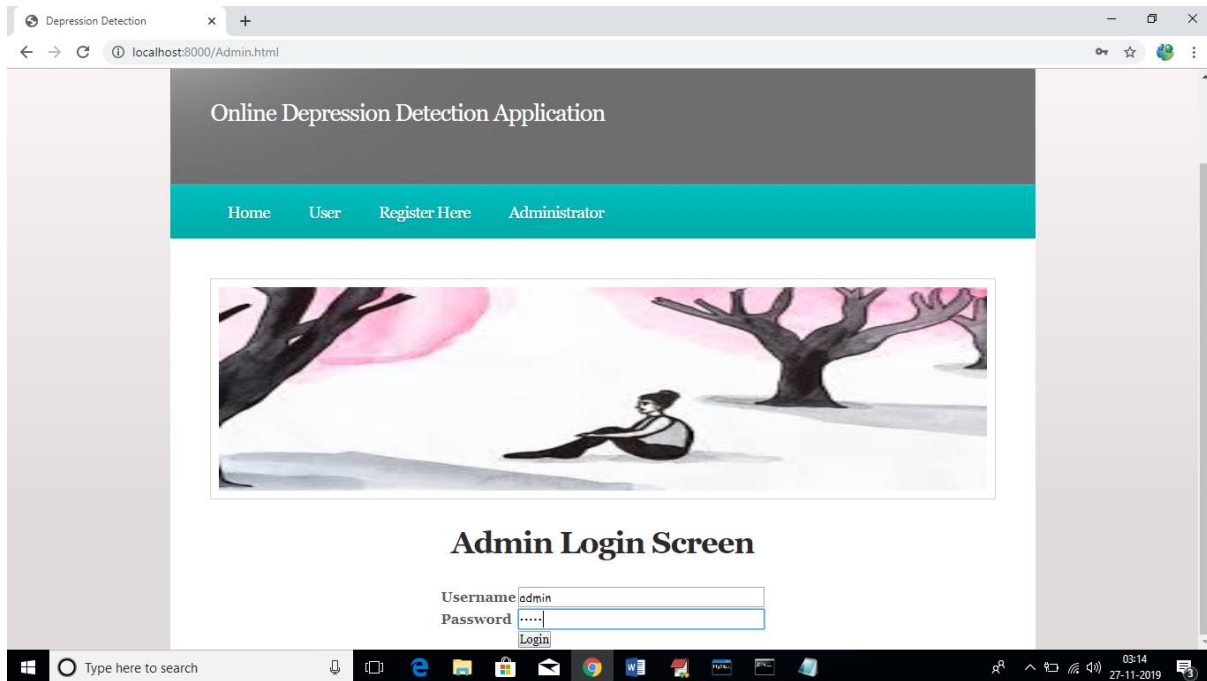
In above screen we can see I am uploading same image and below are is the output screen



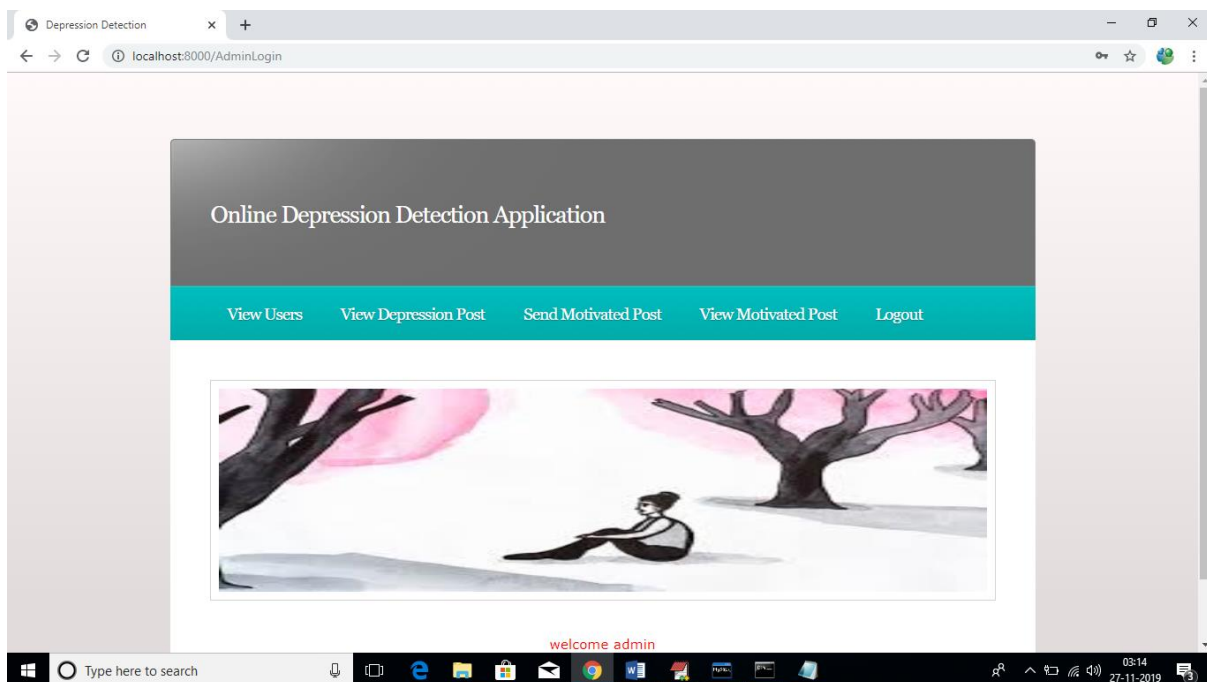
In above screen message is negative and we can say user is depressed. Similarly click on 'View Motivated Post' link to get admin message if he posted otherwise message will mark as 'Pending'



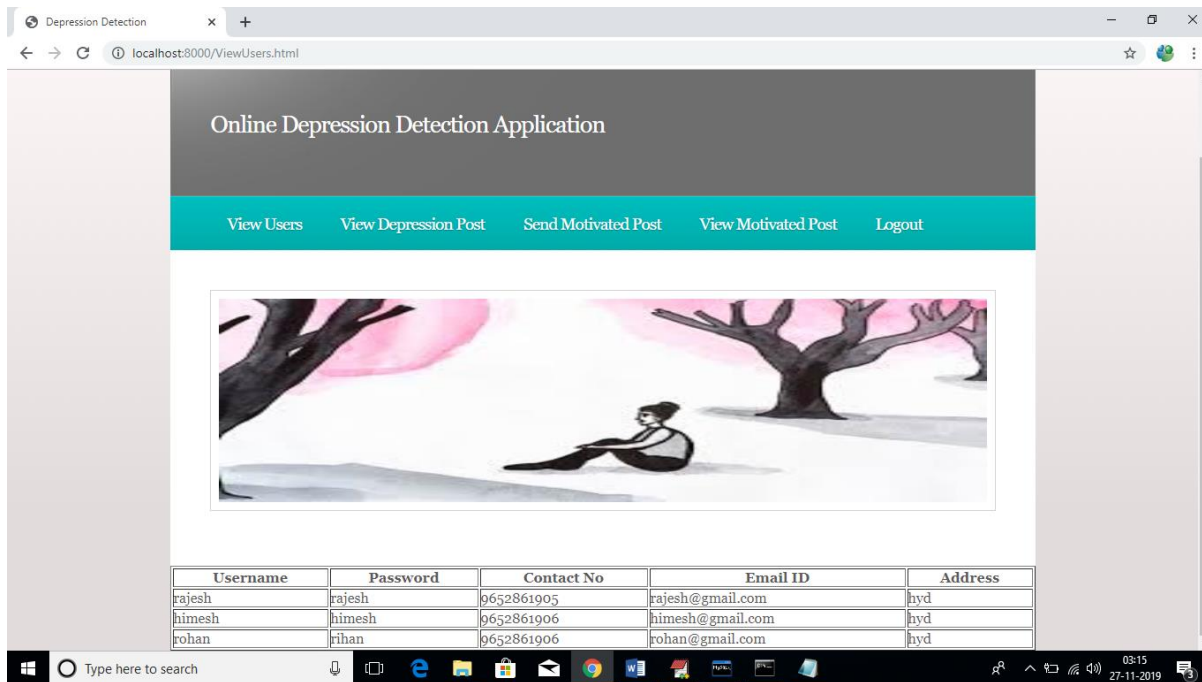
In above screen we can see this user uploaded two files one is recording and other one is image and both file data we can see as post data column and depression result as positive or negative also we can see. Admin has not sent any motivated post so the field is marked as 'Pending'. Now logout and login as 'admin'.



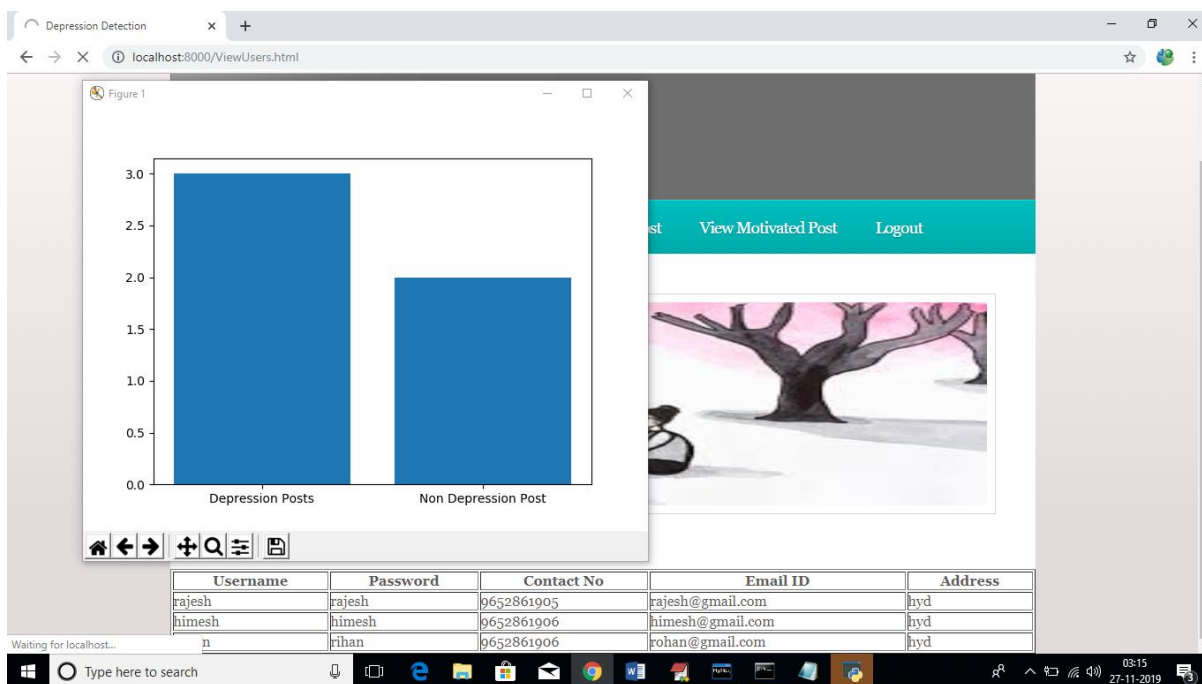
After login we can get below admin screen



In above screen admin can click on 'View Users' to view all users list



Similarly admin can click on 'View Depression Post' to view all post in graph and text format




In above graph we can see total depress and non-depress users, see below screen for all posts from all users

Depression Detection x +

localhost:8000/ViewPosts.html

Online Depression Detection Application

View Users View Depression Post Send Motivated Post View Motivated Post Logout



Username	Post Data	Post Time	Depression	Motivated Post
rajesh	so depressing i m so sick Happy birthday my phone broke	2019-11-27 01:08:33	Negative	dont worry all will be good
himesh	last night movie was good and gave me good relief	2019-11-27 01:16:17	Positive	Pending
rajesh	movie was good	2019-11-27 01:54:38	Positive	Pending
rohan	movie was good	2019-11-27 03:08:06	Positive	Pending
rohan	so depressing i m so sick Happy birthday my phone broke	2019-11-27 03:11:29	Negative	Pending

Type here to search

03:16 27-11-2019


Similarly admin can click on 'Send Motivated Post' link to send motivated messages to users. While sending messages admin has to enter username and post time. Post time he can copy from above screen

Depression Detection x +

localhost:8000/SendMotivatedPost.html

Online Depression Detection Application

View Users View Depression Post Send Motivated Post View Motivated Post Logout



Send Motivated Post Data Screen

Username

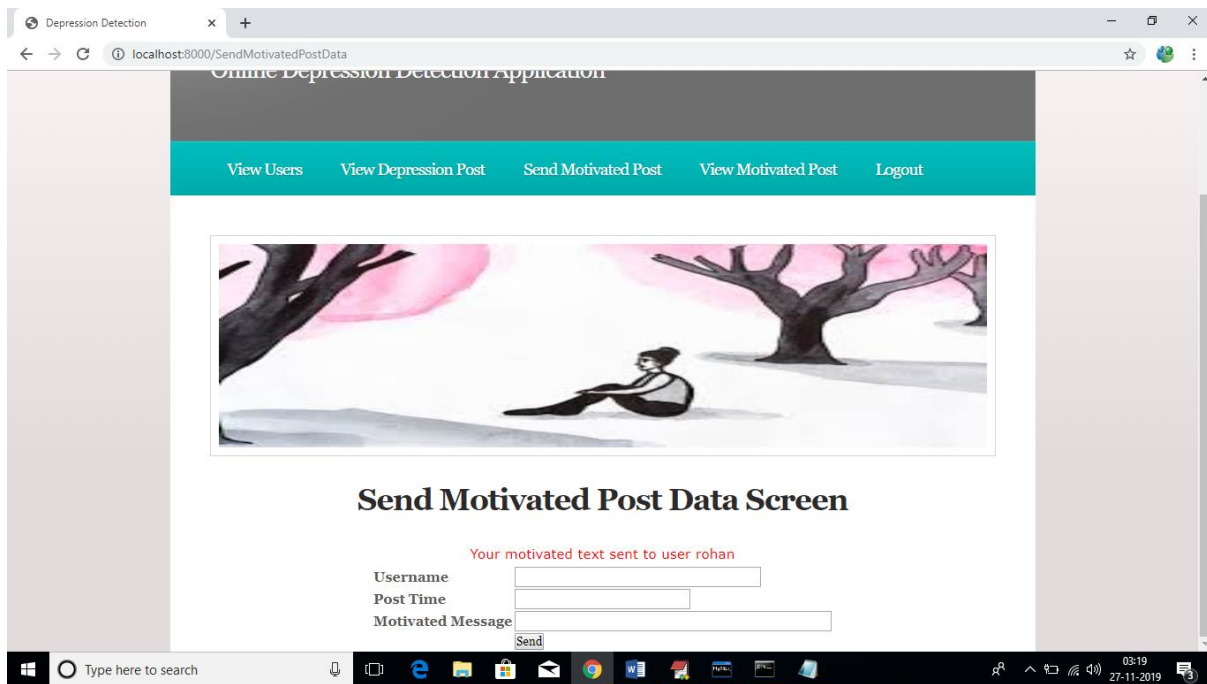
Post Time

Motivated Message

Type here to search

03:18 27-11-2019

After sending motivated messages will get below screen



Now admin can click on 'View Motivated Post' link to view all motivated messages sent by him

CHAPTER - 9

USER MANUAL

This Chapter gives us the detail process that we have to do in the project.

Installing python:

If you're running Windows: the most stable Windows downloads are available from the [Python for Windows](#) page. Screenshot walkthrough, step by step, on bottom of recommendations portion, built for new people learning to **install python on windows PC for the first time**.

1. Visit the [python for windows](#) page.
2. Download python 3.7.0 by clicking the link in step 3.
3. <https://www.python.org/ftp/python/3.7.0/python-3.7.0-webinstall.exe>
4. Begin the download and follow screenshot instructions below.

My story about Uninstalling python 3.7—Pip Uninstall Python -lol -stop
Uninstalling python 3.7 is fast and painless, my goal here is to get my windows machine caught up with my macbook pro...[medium.com](#)

Screenshots from installing python on windows PC Python Release 3.7.0

- Open your executable.
(Or)
- If you're not that techie, navigate to your download folder...
(Or)
- Search your computer for python-3.7.0-webinstall.exe

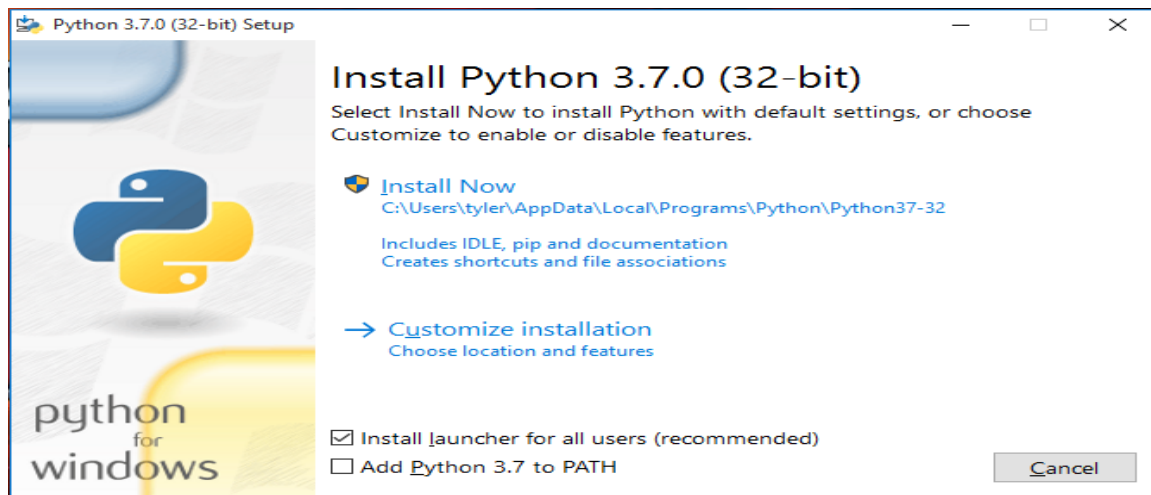


Fig15:Installing Python

Install now option works for me! Check add python 3.7 to path to add it to your system variables.

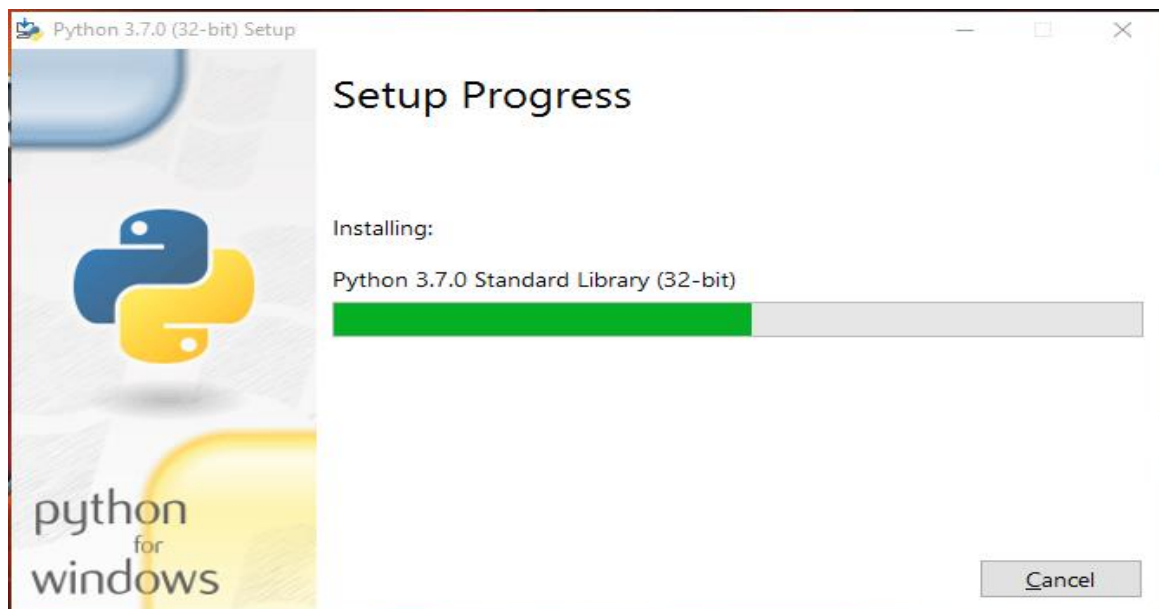


Fig16:Python setup

Setup progress - It describes the progress of the installation of python in windows.

Testing to see if python 3.7.0 installed on your windows machine

1. find your python.exe executable.

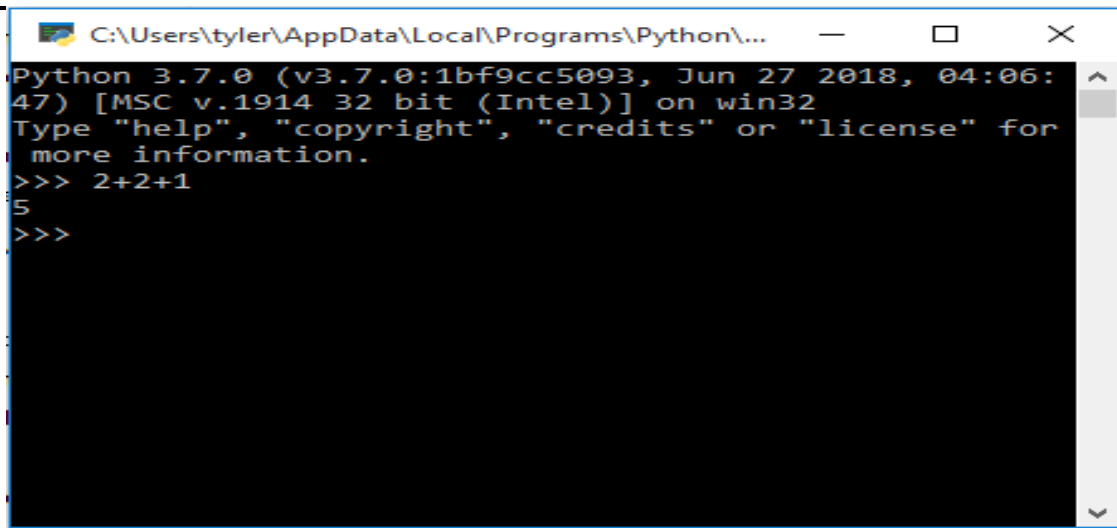


Fig18:Python powershell

Below are software packages we need to install after installing python
install 'tesseract-ocr-setup-3.02.02' (this software u can find inside code folder)
software in C directory and then set path for tesseract-ocr-setup-3.02.02 file using
below environment variable under system tab

Variable name : tesseract

Variable value = C:\Tesseract-OCR\tesseract.exe

Install below softwares using PIP command

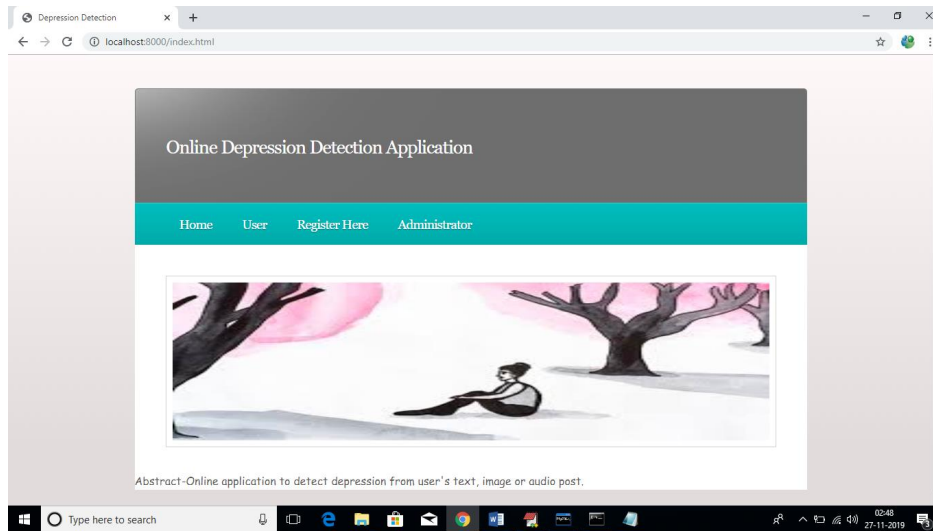
pip install pytesseract

pip install SpeechRecognition

pip install Django

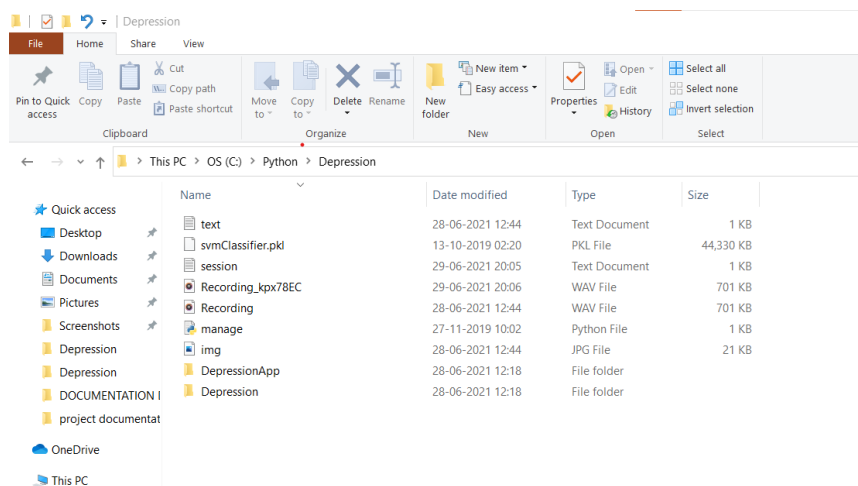
now create database in MYSQL by copying content from 'DB.txt' file and paste in mysql.

Now deploy code in DJANGO and start server and run in browser to get below screen



**For getting the above screen we have to run code in the command prompt.
The below screenshot Shows the execution.**

Step 1:

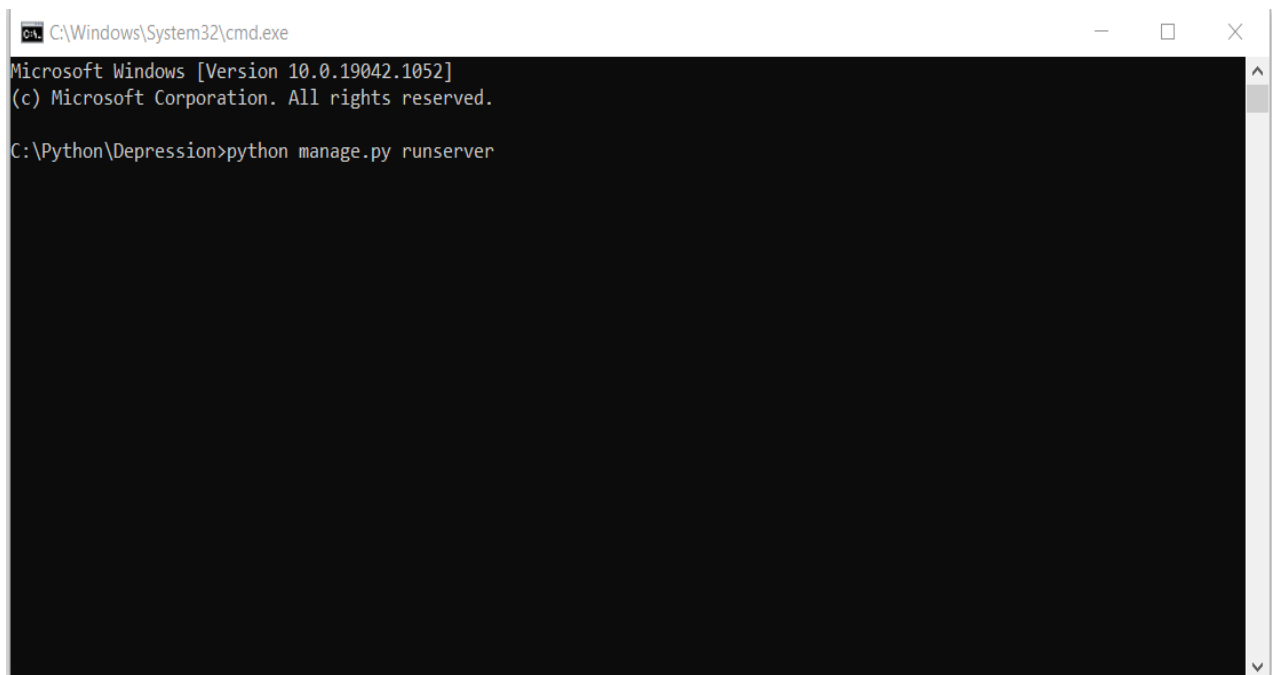


Step 2:

After Opening python folder then in the search bar we search for command prompt then python command prompt is opened for the execution as shown in fig

Step 3:

After opening to run the all over code we will give the below command to see whether our program executes successfully.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Python\Depression>python manage.py runserver
```


Step 4:

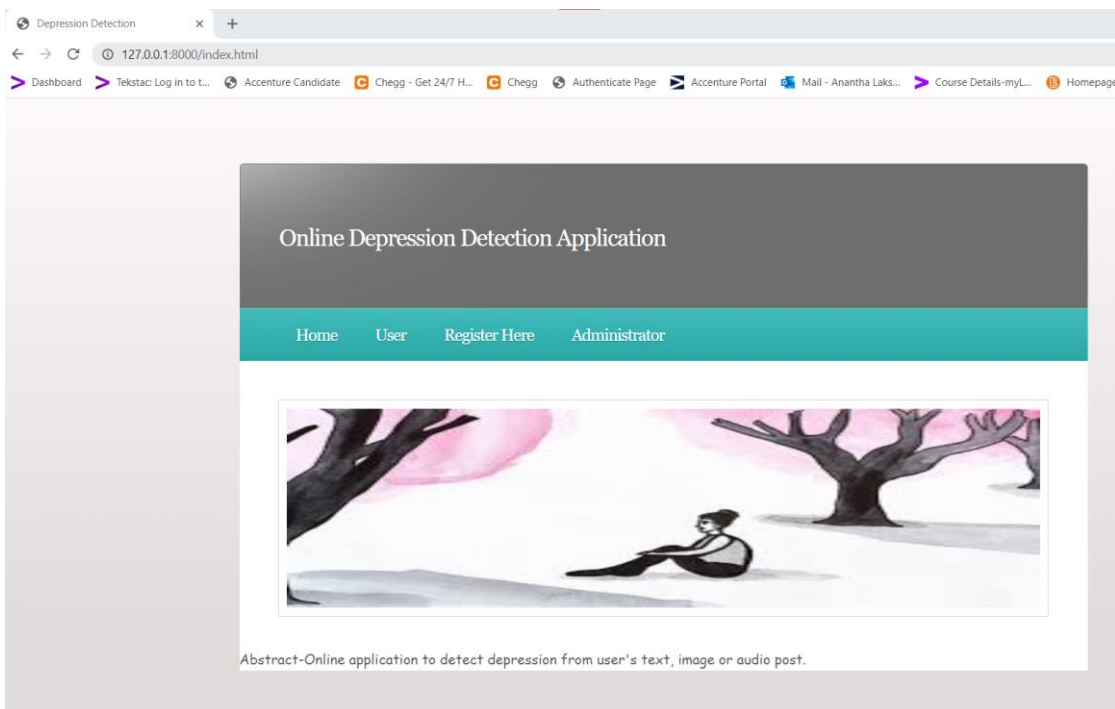
After successful execution the screens looks as follows:

```
C:\Windows\System32\cmd.exe - python manage.py runserver
Warning: sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23. Please import this functionality directly from joblib, which can be installed with: pip install joblib. If this warning is raised when loading pickled models, you may need to re-serialize those models with scikit-learn 0.21+.
warnings.warn(msg, category=FutureWarning)
C:\Users\DELL\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator TfidfTransformer from version 0.21.3 when using version 0.22.2.post1. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(msg, category=UserWarning)
C:\Users\DELL\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator TfidfVectorizer from version 0.21.3 when using version 0.22.2.post1. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(msg, category=UserWarning)
C:\Users\DELL\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator MultinomialNB from version 0.21.3 when using version 0.22.2.post1. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(msg, category=UserWarning)
C:\Users\DELL\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\base.py:318: UserWarning: Trying to unpickle estimator Pipeline from version 0.21.3 when using version 0.22.2.post1. This might lead to breaking code or invalid results. Use at your own risk.
  warnings.warn(msg, category=UserWarning)
System check identified no issues (0 silenced).
You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
July 14, 2021 - 19:28:13
Django version 2.1.7, using settings 'Depression.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Step 5:

After running successful we obtain a url which we have copy and open in the new window.

[url:http://127.0.0.1:8000/index.html](http://127.0.0.1:8000/index.html)



Finally we can obtain our Online Depression Detection Application which looks as the above screen.

CHAPTER - 10

CONCLUSION

The chapter gives brief conclusion about the project.

Conclusion

- In this paper, we tried to identify the presence of depression in Reddit social media; and searched for affective performance increase solutions of depression detection. We characterized a closer connection between depression and a language usage by applying NLP and text classification techniques.
- We identified a lexicon of words more common among the depressed accounts. According to our findings, the language predictors of depression contained the words related to preoccupation with themselves, feelings of sadness, anxiety, anger, hostility or suicidal thoughts, with a greater emphasis on the present and future

Challenges

- Understanding the connections of SQLite Database is a tricky part and confusing when dealing with multiple tables within a database.
- Making exact orientation API design levels was a difficult task as there are many types of devices like desktop, tablet, mobile with varying screen size and resolutions.
- Learning different technologies and frameworks with little guidance.

FUTURE ENHANCEMENT:

- The best feature among the single feature sets is bigram; with SVM classifier it can detect depression with 80% accuracy and 0.79 F1 score. Considering LIWC and LDA features, LIWC outperformed topic models generated by LDA.

Limitations:

- T. Fabian, "Online depression detection in Proc. Comput. Inf. Syst. Ind. Manage. Appl., Jun. 2008, pp. 165170.

CHAPTER - 10

BIBILOGRAPHY

This chapter gives us the list of all sources we have used for project

BIBLIOGRAPHY:

Code snippets for any
errors

<http://stackoverflow.com/>

Android Development Guide

<https://www.udemy.com/android>

Xml and Layout Guide

<https://www.androidhive.com/>

Connecting to Firebase

Docs

<https://firebase.google.com>

Software Testing

http://en.wikipedia.org/wiki/Software_testing

Manual Testing

http://en.wikipedia.org/wiki/Manual_testing

Performance Testing

http://en.wikipedia.org/wiki/Software_performance_testing