**Project Report**

# **The Psychiatrist** (Real-Time Sentiment Analysis of tweets on Twitter for Identifying People with Suicidal Tendencies with AI)


**Submitted by**

**Ammara Khan** (BSCS-Fa17/BSCS/003-C)

**Zubair Mustafa** (BSCS- Fa17/BSCS/081-C)


Session Fall 2017-2021


**Supervised by**

**Jawad Hassan**




**Department of Computer Science**

**Lahore Garrison University**

**Lahore**

# **The Psychiatrist** (Real-Time Sentiment Analysis of tweets on Twitter for Identifying People with Suicidal Tendencies with AI)

A project submitted to the

Department of Computer Science

In

Partial Fulfillment of the Requirements for the

Bachelor's Degree in Computer Science

By

**Ammara Khan**

&

**Zubair Mustafa**

**Supervisor**

**Jawad Hassan** _____
Lecturer
Department of Computer Science

**External Examiner** _____

**Chairperson**

**Dr. Muhammad Asif** _____
Head of Department / Assistant Prof.
Department of Computer Science

# COPYRIGHTS

This is to certify that the project titled "**The Psychiatrist** (Real-Time Sentiment Analysis of tweets on Twitter for Identifying People with Suicidal Tendencies with AI)**"** is the genuine work carried out by **Ammara Khan** and **Zubair Mustafa,** students of BSCS of Computer Science Department, Lahore Garrison University, Lahore during the academic year 2017-2021, in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Ammara Khan _____

Zubair Mustafa _____

# DECLARATION

This is to declare that the project entitled "**The Psychiatrist** (Real-Time Sentiment Analysis of tweets on Twitter for Identifying People with Suicidal Tendencies with AI)" is an original work done by undersigned, in partial fulfilment of the requirements for the degree "Bachelor of Science in Computer Science" at Computer Science Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Ammara Khan _____

Zubair Mustafa _____

# ACKNOWLEDGEMENTS

We would like to appreciate everyone who provided us the opportunity to complete this report. Working on this project was not an easy task.

We would like to thank our project supervisor **Mr. Jawad Hassan** for his guidance and support during this whole time, as it kept us motivated to work hard on the project.

We are grateful to the management of **Department of Computer Sciences** who allowed us to work on this project and their guidance throughout the project completion period.

# DEDICATION

We would like to dedicate this work to the **Department of Computer Sciences** and our Supervisor for this project **Mr. Jawad Hassan** because of whom we stayed motivated and were able to work complete the project.

We dedicate our work to these people in hope that some student in the future will work on this project and will make it more advanced and perfect.

# Table of Contents


**CHAPTER-1** ..... 1

**1.1 Aim of Project** ..... 2

**1.2 Report Organization** ..... 2

**CHAPTER-2** ..... 4

**Problem Statement** ..... 4

**CHAPTER-3** ..... 5

**3.1 Purpose** ..... 5

**3.1.1 Document Conventions** ..... 5

**3.1.2 Intended Audience and Reading Suggestions** ..... 6

**3.1.3 Product Scope** ..... 6

**3.2 Overall Description** ..... 7

**3.2.1 Product Perspective** ..... 7

**3.2.2 Product Functions** ..... 7

**3.2.3 User Classes and Characteristics** ..... 7

**3.2.4 Operating Environment** ..... 8

**3.2.5 Design and Implementation Constraints** ..... 8

**3.2.6 User Documentation** ..... 9

**3.2.7 Assumptions and Dependencies** ..... 9

**3.3 Project Requirement** ..... 9

**3.4 External Interface Requirements** ..... 10

**3.4.1 User Interfaces** ..... 10

**3.4.2 Hardware Interfaces** ..... 11

**3.4.3 Software Interfaces** ..... 11

**3.4.4 Communications Interfaces** ..... 11

**3.5 System Features** ..... 11

**3.5.1 Pull Tweets** ..... 11

**3.5.1.1 Description and Priority** ..... 12

**3.5.1.2 Stimulus/Response Sequences** ..... 12

**3.6 Functional Requirements** ..... 12

**3.6.1 Sentiment Analysis** ..... 12

**3.6.2 Description and Priority** ..... 12
**3.6.3 Stimulus/Response Sequences** ..... 12
**3.6.4 Functional Requirements** ..... 12
**3.7 Display Result** ..... 13
**3.7.1 Description and Priority** ..... 13
**3.7.2 Stimulus/Response Sequences** ..... 13
**3.7.3 Functional Requirements** ..... 13
**3.8 Other Non-functional Requirements** ..... 13
**3.8.1 Performance Requirements** ..... 13
**3.8.2 Safety Requirements** ..... 13
**3.8.3 Security Requirements** ..... 13
**3.8.4 Software Quality Attributes** ..... 13
**3.8.5 Business Rules** ..... 14
**3.9 Literature Review** ..... 14
**3.9.1 Existing System** ..... 14
**3.9.2 Proposed System** ..... 14
**CHAPTER-4** ..... 15
**4.1. Training:** ..... 15
**4.2. Testing:** ..... 15
**4.2.1. Backend Application** ..... 16
**4.2.2. Frontend Application** ..... 16
**4.2.3. Database/Storage** ..... 17
**4.3. Approach, Tools, Modules** ..... 17
**4.3.1. Approach** ..... 17
**4.3.2. Tools** ..... 17
**4.3.3. Modules:** ..... 18
**4.4. Training Results:** ..... 18
**CHAPTER-5** ..... 19
**5.1 SYSTEM ARCHIECTURE** ..... 19
**5.1.1 Architecture Design Approach** ..... 19
**5.1.2 Architecture Design** ..... 20
**5.1.3 Subsystem Architecture** ..... 21
**5.2 DETAILED SYSTEM DESING** ..... 22

**5.2.1 Classification** ..... 22
**5.2.2 Definition** ..... 23
**5.2.3 Responsibilities** ..... 23
**5.2.4 Constraints** ..... 23
**5.2.5 Composition** ..... 23
**5.2.6 Uses/Interactions** ..... 24
**5.2.7 Resources** ..... 24
**5.2.8 Processing** ..... 24
**5.2.9 Interface/Exports** ..... 24
**5.2.10 Detailed Subsystem Design** ..... 24
**CHAPTRER-6** ..... 28
**6.1 IMPLEMENTATION** ..... 28
**6.1.1. Implementation** ..... 28
**6.2. GUI** ..... 30
**6.2.1. User Panel:** ..... 30
**6.2.2. Admin Panel:** ..... 37
**CHAPTER-7** ..... 43
**7.1. Results** ..... 43
**7.2. Discussions** ..... 44
**CHAPTER-8** ..... 45
**8.1. Conclusion** ..... 45
**8.2. Future Work** ..... 45
**References** ..... 46

# List of Tables

3.1: Document Conventions ---------- 5

3.2: Hardware Requirements ---------- 9

3.3: Software Requirements ---------- 10

4.1: Training Models Results ---------- 18

# List of Figures

5.1.1: System Architecture Design ---------- 20
5.1.2: Sub-system Architecture Design of Backend Subsystem ---------- 21
5.1.3: Sub-system Data Flow Diagram of Backend Subsystem ---------- 21
5.1.4: Sub-system Architecture Design of Frontend Subsystem ---------- 22
5.1.5: Sub-system Data Flow Diagram of Frontend Subsystem ---------- 22
5.2.1: Use case Diagram ---------- 25
5.2.2: Activity Diagram ---------- 26
5.2.3: Data Flow Diagram ---------- 27
5.2.4: Database Diagram ---------- 27
6.1: How Twitter API works ---------- 28
6.2.1: Enter Development Mode ---------- 29
6.2.2: Flask APP ---------- 29
6.2.3: Run APP ---------- 29
6.2.4: APP URL---------- 29
6.2.5: Web-Frontend URL Paste ---------- 30
6.2.6: Home Page ---------- 31
6.2.7: Contact Page ---------- 32
6.2.8: Login Page ---------- 33
6.2.9: Signup Page ---------- 34
6.2.10: Cases Page ---------- 35
6.2.11: Update Case Page ---------- 36
6.2.12: Admin Login Page ---------- 37
6.2.13: Dashboard Page ---------- 38
6.2.14: User Management Page ---------- 39
6.2.15: User Update Page ---------- 40
6.2.16: Add User Page ---------- 41

# List of Abbreviation

| |
|---|
| HTML - Hyper Text Markup Language |
| JS - JavaScript Language |
| Materialize CSS - Materialize Cascading Style Sheet |
| ABC - Ada Boost Classifier |
| RFC – Random Forest Classifier |
| SVM – Support Vector Machine |
| SRS – Software requirement specification |

# Abstract

It has been widely observed that people with suicidal tendencies and people who commit suicide often discuss it on the social media and can be saved if identified at the right time. Any and all the solutions provided so far only identify the suicidal tweets but not the users. The proposed solution identifies the people with suicidal tendencies and suicidal people on the bases of the sentiment analysis of their tweets and shows the results on the web-based frontend. The proposed solution used Twitter API v2 to retrieve tweets, python-based backend used four algorithms *Ada Boost Classifier (ABC)* with *Random Forest Classifier (RFC) and Gradient Boosting Classifier for n-gram (1,3) and n-gram (1,2) respectively, Support Vector Machine (SVM) and Ada Boost Classifier (ABC)* with *Decision Tree Classifier*. With accuracies 86%, 88.2%, 85.6%, 88.6%,90% and 80.6%% respectively**.** SVM model was used for tweets prediction as it had the highest accuracy among all algorithms**.** It was observed that the proposed solution's accuracy was **5-10%** more than the other observed solutions. The frontend and database of the solution is made using *FLASK, HTML, JS and Materialize CSS*. With its accuracy the web application can be used to identify people with suicidal tendencies and suicidal people and they can be provided help on time. It is hoped that this will attract attention and more work will be done to make this application better and more perfect and that the project inspired people to work on problems like this.

# CHAPTER-1

# Introduction

This report provides the documentation for the Final Year Project named “The Psychiatrist”, in this documentation, the project’s background, working, methodology behind its working will be explained.

The project has two parts, the frontend and the backend. The backend application retrieves tweets from twitter using Twitter API, applies sentiment analysis on the tweets one at a time. If the tweet is identified as suicidal, it will be saved in the database in the backend, else the tweet is discarded and next tweet is taken for analysis. The frontend requests the database for the identified tweets data, retrieves it and displays it in a tabular form. It also lets the user to change the identified case’s status as the case progresses between Identified, In Progress and Solved/Closed and the status is then saved in the database in the backend. The frontend also provides a link to the identified user’s profile and the identified tweet.

The project aims to identify suicidal people or people who show signs of being suicidal on the basis of their tweets so that they can be provided help in time. All the previous solutions that were observed only identified the tweets but not the people. The project was made solely to provide a way to help the people who need it.

This project can be used by:

- **Police/Government Officials:**

  This is the primary category of users that is targeted as they are the first responders of a potential suicide case. They will be able to use this project to get user ID of the suicidal user and have the concerned authorities involved to provide help to the identified person.

- **Health Researchers:**

  Many health researches carry out researches against the rate of suicide, health conditions of a country, cause of suicide etc. The researchers would be able to use this

project in order to quickly perform analytics on the data pulled from an actual social network.

- **Students:**

This is the tertiary class to be targeted because if they find this project interesting, they may be able to modify or enhance the features to make this project more helpful to the primary and secondary consumers.

This chapter briefly explained working and the purpose behind the project. It is hoped that this project can prove helpful to people. The next chapters explain the Problem Definition, SRS, Methodology and Design and Architecture for the project in detail.

## 1.1 Aim of Project

The main aim of the project is to ***decrease the rate of suicides in our country specially in students.***

## 1.2 Report Organization

This report is organized into divisions, with each chapter devoted to a different phase of the project.

In this chapter (chapter-1), introduction of project is briefly described as a non-user can easily understand about this project. Below is the detail of other chapters:

- *Chapter **2***: Describes the problem definition according to project that why this project is made.
- *Chapter **3***: Discusses the functional and non-functional needs, as well as other general considerations, in this description of the Software Requirement Specification.
- *Chapter **4***: Describes the methods of application.
- *Chapter **5***: Complete Software Architecture Design, step by step designs approach, UML diagrams, E-R diagram and other diagrams is described.

- *Chapter **6***: Showcases all of the project's test plans, implementation and describes the tools and techniques used to develop the software.
- *Chapter **7***: Results are discussed after testing is performed.
- *Chapter **8***: Concluded work and future work of project is discussed in this chapter.

# CHAPTER-2

# Problem Definition

## Problem Statement

According to a lot of suicide studies, over 800,000 individuals commit themselves each year, and spotting suicidal people remains a difficult task. As the usage of social media has grown, we've seen people openly discuss their suicide plans or attempts on these platforms [1].

Suicides among teenagers and young adults continue to be a major issue. For children, adolescents, and young people aged 15 to 24, suicide is the second highest cause of mortality. The majority of children and adolescents who attempt suicide suffer from a serious mental illness, most commonly depression.

Many people tweet which shows that they are in stress or depression which leads to suicide. So, by this project, we will reduce suicides in our country.

# CHAPTER-3

# Software Requirement Specification

This section gives the description and overview of everything included in the Software Requirement Specifications (SRS) of the project. Furthermore, purpose for this document is also covered in the introduction.

## 3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the **"The Psychiatrist"** (Real-Time Sentiment Analysis of tweets on Twitter for Identifying People with Suicidal Tendencies with AI) web-based application. It will also explain the constraints and interface required by the application for complete execution.

The purpose for developing this web-based application is to provide the concerned authorities, especially law enforcement with such an application that will help them identify Twitter users that have suicidal tendencies so that they can be provided help and medical assistance in time.

### 3.1.1 Document Conventions

Following are the conventions observed in this document:

Table 3.1: Document Conventions

| **Font Style** | **Descriptions** |
|---|---|
| **Bold** | The technology stack |
| *Italic* | Reference to sections or piece of section |
| ***Bold & Italic*** | Priority of System Features |

### 3.1.2 Intended Audience and Reading Suggestions

This web-based application can be used by anyone who is willing to analyze the sentiments of people on social media and figure out if they have suicidal tendencies. The intended audience for this project includes primarily the Computer Sciences Department's faculty and secondarily the law enforcement agencies or medical research institutions. This document is expected to be read by the development team, alpha and beta testers. Others involved in the project anyhow need to read the document as such:

- Overall Description
- External Interface Requirements
- System Features
- Non-Functional Requirements

### 3.1.3 Product Scope

The project uses **Twitter API** which will be used to retrieve real-time tweets from Twitter. These tweets are then subjected to an analyzer script written in **Python** language. To apply machine learning/artificial intelligence algorithms on the tweets retrieved from Twitter, **IBM Watson** will be used. This will aid the sentiment analysis and in identifying if the user is suicidal or not. The front-end of the web-application is in **HTML, CSS** and **JavaScript** which displays the identified suicidal users in the past and the currently identified ones to keep a record.

This product is expected to be beneficial to the First Responders (Law Enforcement Agencies/Government Officials/Health Officers) in identifying suicidal people on Twitter and providing them required help in time. The product is also expected to become a helpful platform for reducing suicide rate as the victims may be identified before it's too late.

## 3.2 Overall Description

This section is intended to provide an overview of the features and the operational functions of the application. It will also declare the stakeholders of the application. Finally, the assumptions and dependencies for the software will be explained.

### 3.2.1 Product Perspective

Our project is new and does not utilize or depend on any of the features or capabilities of any of the applications similar to it. All the related projects that we observed were limited to finding if the tweet was suicidal or not and none of the projects identified the user that posted those suicidal tweets. Our project will identify the user IDs by analyzing the tweets posted in real-time.

### 3.2.2 Product Functions

The project will perform the following functions:

- Pull Tweets in near real-time
- Perform Sentiment Analysis
- Display Case
- Update Case Status
- Delete Case

### 3.2.3 User Classes and Characteristics

This project can be used by:

**Police/Government Officials:**

This is the primary category of users that is targeted as they are the first responders of a potential suicide case. They will be able to use this project to get user ID of the suicidal user and have the concerned authorities involved to provide help to the identified person.

**Health Researchers:**

Many health researches carry out researches against the rate of suicide, health conditions of a country, cause of suicide etc. The researchers would be able to use this project in order to quickly perform analytics on the data pulled from an actual social network.

**Students:**

This is the tertiary class to be targeted because if they find this project interesting, they may be able to modify or enhance the features to make this project more helpful to the primary and secondary consumers.

### 3.2.4 Operating Environment

The project will operate with the following components:

As the web-based application under development is platform independent, it will be running on a desktop machine that can have either a Linux or a Windows operating system. The software requirement for the project is to have Python language framework installed on the machine where the application will run. There are no separate hardware components required for this project.

### 3.2.5 Design and Implementation Constraints

Following are the issues that might limit the application's use:

- Twitter API access is hard to get, it might take a while to get access to the Twitter's development API key. If you are not granted access to the API, you can't use this application
- Availability to IBM Watson is also necessary as it will be used for sentiment analysis of the tweets

### 3.2.6 User Documentation

For user documentation and information, please consult *Section 3: External Interface Requirements* of this document.

### 3.2.7 Assumptions and Dependencies

It is assumed that the web-based application will work correctly with the availability of the following tools and equipment:

**Twitter API:**

The application's working depends mainly on the access to the Twitter API. If the access is not granted, the application will not be able to pull tweets to apply sentiment analysis on them

**Python:**

Python language will be used to develop both front and backend of the application. Related python modules will be used for each function.

**Internet:**

The availability of Internet is required for the application's web-based frontend and to pull Tweets from Twitter.

## 3.3 Project Requirement

There are two requirements needed for this project

- Hardware Requirements
- Software Requirements

**Hardware Requirements**

Table 3.2: Hardware Requirements

| | |
|---|---|
| *Laptop* | Any laptop of good company |

| | |
|---|---|
| *Browser* | Microsoft Edge/Google Chrome (recommended) |
| *Processor* | *Quad Core 1.7 GHz* |
| *RAM* | *4 GB (Minimum)* |
| *Hard Disk* | *4 GB (Minimum)* |
| *Operating System* | *Windows 10 (For Best working)* |
| *System Type* | *32-bit Operating System (minimum)* |

**Software Requirements**

Table 3.3: Software Requirements

| | |
|---|---|
| *Data Base Server* | FLASK SQLAlchemy |
| *Development Tools* | *VS Code/Sublime Text, Terminal, Twitter API, FLASK framework, Materialize* |
| *Programming Language* | *Python* |
| *Documentation of Application* | *Microsoft Word* |

## 3.4 External Interface Requirements

This section contains detailed information of all of the interfaces in the project.

### 3.4.1 User Interfaces

It is assumed that the primary users of this application are not technical, so the user interface is kept very simple and understandable by non-technical users. The user only has to run the

application and access in the browser. The sentiment analysis will be done in the background and the results will be showed on the web-based frontend of the application.

### 3.4.2 Hardware Interfaces

Since the software does not require any designated hardware, it does not have any direct hardware interfaces. The internet is managed by the Wi-Fi router of the desktop machine hosting the application.

### 3.4.3 Software Interfaces

The Psychiatrist application runs on either a Linux or a Windows machine. The application front-end can be accessed primarily by Google Chrome and secondarily by any equivalent web browser. The backend of the application is built using Python for tweet retrieval from Twitter API and implementing the algorithms applied for sentiment analysis of tweets. The frontend is primarily built using HTML, CSS JavaScript and Materialize.

### 3.4.4 Communications Interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the primary user and is therefore handled at the backend of the web-based application.

## 3.5 System Features

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features. The features included in the system can be viewed in *Section 2: Overall Description* of the document. Below, they will be discussed in detail:

### 3.5.1 Pull Tweets

Pulling tweets from Twitter.

#### 3.5.1.1 Description and Priority

This feature will be used for pulling real-time tweets from Twitter through its developer API. This is a ***HIGH*** feature as these are the tweets that will be used for analysis.

#### 3.5.1.2 Stimulus/Response Sequences

When user runs the application, the python-based backend will connect to Twitter through the API and start pulling real-time tweets from Twitter.

## 3.6 Functional Requirements

The user must have access to Twitter API and Secret keys as it is impossible to pull tweets from Twitter without them.

### 3.6.1 Sentiment Analysis

Applying sentiment analysis on retrieved tweets.

### 3.6.2 Description and Priority

This feature will be used to apply sentiment analysis on the pulled tweets to determine whether they are suicidal or not. It is a ***HIGH*** priority feature as the quality of results depends on the analysis performed on the Tweets.

### 3.6.3 Stimulus/Response Sequences

After the tweets are pulled, they will be sent to the analysis model by the backend application where algorithms for sentiment analysis will be applied to determine if the tweet is normal or suicidal. If the tweet is suicidal, it will be sent to the database of application for saving and display.

### 3.6.4 Functional Requirements

The application user must have an account and access to related services created on IBM Cloud because without this it is not possible to apply sentiment analysis on tweets.

## 3.7 Display Result

Displaying Tweet text, User Profile link, Tweet link, Case status, case update option and case deletion option on the frontend.

### 3.7.1 Description and Priority

This feature will be used for displaying the identified Twitter user's ID on the web-based frontend. This is a ***HIGH*** priority as the quality and result of the application is dependent on it.

### 3.7.2 Stimulus/Response Sequences

If the analyzed tweet is determined as suicidal, the user ID that posted the tweet is pulled through the API and showed at the web-based frontend of the application.

### 3.7.3 Functional Requirements

The application user must have access to the internet so that the web-based frontend can work properly.

## 3.8 Other Non-functional Requirements

This section contains all the non-functional requirements of the system.

### 3.8.1 Performance Requirements

None

### 3.8.2 Safety Requirements

For data protection use any of the cloud storage and backups to prevent any data loss.

### 3.8.3 Security Requirements

Only authorized personal can view suicidal cases displayed

### 3.8.4 Software Quality Attributes

The project is reliable and portable as it can be used on any computer running any Operating System that supports Python language framework and it will give results using multiple

algorithms. It will help to understand different sentiments of users through analyzing their tweets.

### 3.8.5 Business Rules

The web-based application will be used by the primary users i-e the Police/Government who are often the first responders of suicide cases. Otherwise, it can be used by students or health researchers who wish to modify or add more features to this application.

## 3.9 Literature Review

### 3.9.1 Existing System

Suicide prevention research has mostly focused on recognising suicidal thinking or ideation, as well as detecting suicidal postings. However, no major study has been done on the identification of suicidal profiles in particular. As a result, by recognising suicidal profiles on Twitter, we want to contribute to the research on understanding suicide communication in social networks.

Another study looked at online sites like Reddit and used topic analysis and linguistic characteristics to spot behavioral changes and mental health concerns like suicide thoughts, emphasizing the dangers of ostensibly beneficial posts in such communities.

### 3.9.2 Proposed System

We present a technique for detecting suicidal tendencies in this paper. To begin, we use the most easily accessible data to examine a number of accounts from the social networking site Twitter. Then, to discriminate between suicidal and non-suicidal profiles, we use a set of criteria. These characteristics can be derived directly from the user profile or inferred implicitly using a number of data mining tools and methodologies. In this part, we'll look into emotional characteristics and sentiment analysis, which can provide insight into a suicidal profile's psychological condition.

# CHAPTER-4

# Methodology

This chapter gives the description and overview of the methodology used in the project. Furthermore, purpose of all the modules and tools used in the project is also provided.

The project used supervised learning approach to classify tweets as suicidal/1 or non-suicidal/0. AdaBoost combined with Random Forest Classifier, Gradient Boosting Classifier, SVM and AdaBoost combined with Decision Tree Classifier were used for training because these algorithms had the highest accuracy and are good for text classification.

All tools, techniques, modules used in this project are provided in Sub-Section Approach, Tools and Modules.

## 4.1. Training:

Model training included three steps given below:

1. **Preprocessing:**

   During this step the tweets were cleaned by converting them to lowercase, removing punctuations, tokenization, stop words removal using NLTK and Lemmatization/stemming.

2. **Train-Test Split:**

   Dataset consisting of 10,728 tweets was split into training and testing set, N-gram technique was applied on the split sets.

3. **Training:**

   The dataset was trained using two models, AdaBoost combined with Random Forest Classifier and SVM model for classification. The dataset was further tested using 3-Fold Validation on both models. Both models were saved separately.

## 4.2. Testing:

The application consisted of 2 sub applications and the storage/database.

- Backend Application for tweet identification
- Frontend Application for displaying identified cases.

- Database for storing identified tweet's information.

### 4.2.1. Backend Application:

The system's backend application used Twitter API version 2 to retrieve the real-time tweets information which by default are in json format. The tweet information included **Author ID** (user's id), **Tweet ID** and the **Tweet Tex**t itself and was saved in a list form. The list was looped using **FOR Loop** to analyze and predict sentiment of each tweet. The model took each tweet and identified it as either **0 (non-Suicidal)** or **1(Suicidal).** The tweet that was identified as suicidal, it's information as well as its sentiment was saved in the database. By default, the status of the case was saved as **Active**, which could be changed through web-based frontend as the case progressed. After a case was finished, it could be deleted from the database.

### 4.2.2. Frontend Application:

The system's frontend application consisted of two panels.

Admin Panel

- Dashboard
- Users
- Update User
- Add User

User Panel

- Home
- Contact
- Cases

The cases page, protected through a login process displayed the identified cases. It showed the information in tabular form. The information displayed on the page, taken from the database using JavaScript included

- ID
- Tweet text
- Sentiment

- User Profile Link
- Tweet Link
- Status
- Update Case
- Delete Case

A user could go to the home page by default, contact page and cases page to view identified cases after logging in and return to the home page on clicking the logout button.

Both frontend and backend work parallelly through multithreading.

### 4.2.3. Database/Storage:

The database was saved in SQLite format of SQLAlchemy module with two tables.

- Users table to store the credentials and information of application users.
- Results table to store predicted tweets information.

## 4.3. Approach, Tools, Modules

Following are the approach, tools and modules used in the project.

### 4.3.1. Approach:

The project used Machine Learning approach called Supervised Learning for classification that takes example input/output pairs to learn and after learning predicts the output for new input.

### 4.3.2. Tools:

Following tools were used in project development

- Python 3.8 - used to create the project. For training and real time analysis.
- FLASK – Python framework used for web development
- Materialize – for responsive frontend framework development (like Bootstrap)
- YAML – for project configurations
- HTML, CSS, JavaScript – for frontend development
- Twitter API access – for retrieving real-time tweets from twitter
- Sublime Text/Visual Studio Code – for project development

### 4.3.3. Modules:

Following python modules were used in project development:

- Pandas – for training dataset file processing.
- NLTK – two sub-modules used in training process for text tokenization i-e. separating text into smaller units which are called tokens and removing common words like (is, am, are etc.).
- Text blob – used for text lemmatization/stemming to convert a word into its root word like converting killing, killed to kill etc.
- Sklearn – to split dataset into training and testing sets, training the dataset using SVM, Random Forest Classifier and Ada Boost Classifier models, get the models accuracy and reports and cross validation.
- Joblib – for saving and loading the models.
- Requests – for making HTTP GET requests to retrieve tweets and related information from twitter using the URL provided.
- Time – to stop retrieving tweets and rest for a specified time
- Threading – for multithreading so that the frontend web application and backend python application run parallelly.
- SQLAlchemy – FLASK module for database management.

All the modules required and not provided by python by default are added in the **"requirements"** text file provided with the project. User only needs to install the mentioned file through terminal.

## 4.4. Training Results:

Table 4.1: Training Models Results

| | ABC/RF n-grams (1,3) | GB n-grams (1,3) | ABC/RF n-grams (1,2) | GB n-grams (1,2) | SVM | ABC/DT n-grams (1,2) |
|---|---|---|---|---|---|---|
| **Accuracy** | 86.2% | 88.2% | 85.6% | 86% | 90% | 80.6% |
| **3-Fold Cross Validation** | 86.2% | 88.2% | 85.6% | 85.6% | 90% | 80.6% |

# CHAPTER-5

# Detailed Design and Architecture

## 5.1 SYSTEM ARCHIECTURE

This chapter provides the detailed design and architecture of the project. Furthermore, in-depth details are also provided.

The system was decomposed into two sub-systems for easy maintenance and working. First sub-system would run in the background/backend to classify tweets and save data in the database. The second subsystem would run on the foreground/frontend for user to work with.

The software as a whole would retrieve real-time tweets from Twitter through API, provide sentiment analysis of each tweet. Save the tweet identified as suicidal in the database including the related information. Display the records on the web-based frontend. Update the case as it progresses and delete the case from database when it is closed to help application work accordingly. The sub-system running in the backend would take real-time tweets from twitter using API. Run analysis on each tweet to classify them as suicidal or non-suicidal. If a suicidal tweet is predicted, the related information would be saved in the database.

### 5.1.1 Architecture Design Approach

This section provides the architecture design approach applied in the project.

The Architecture Design Approach used in the project is the **Call and Return Architecture**. This architecture approach was used because it is used to create a program that is easy to scale and modify. It has multiple sub styles. The one used in this project is **Main program or Subprograms Architecture**. The main program structure decomposes into number of subprograms or function into a control hierarchy. Main program contains number of subprograms that can invoke other components [2].

### 5.1.2 Architecture Design

This section explains the architecture design of the project.

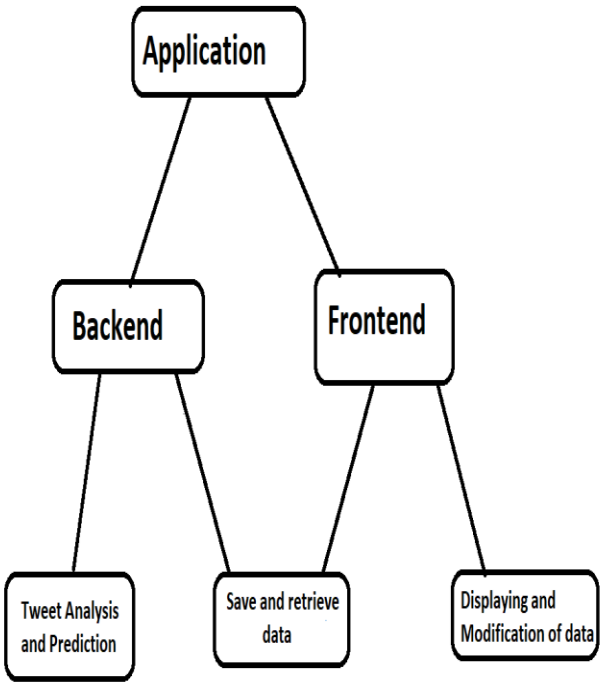


***Figure 5:1:1 System Architecture Design***

The application/system was divided into two sub-systems. The backend application/sub-system for tweet prediction and saving in the database, all happening in the background. The frontend application/subsystem for displaying predicted results and for the user to modify record as needed. As the backend predicts/identifies a user as suicidal, it saves the related information in the database. The frontend accesses the saved information from the database and displays it on the web-based frontend of the system. The both sub-systems work parallelly through multithreading.

### 5.1.3 Subsystem Architecture

This section explains the sub-systems architecture design of the project.

### Backend Application Sub-System

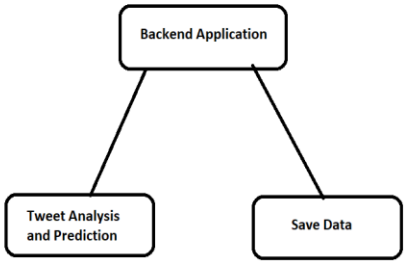


***Figure 5:1:2 Sub-system Architecture Design of Backend Application***

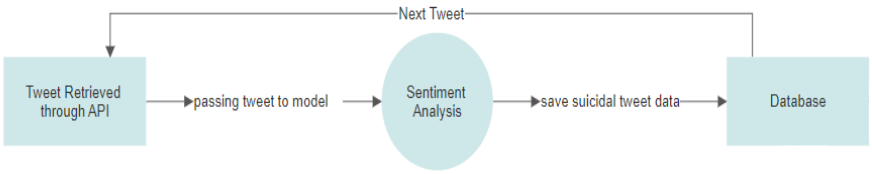


***Figure 5:1:30:1 Sub-system Data Flow Diagram of Backend Application***

This sub-system predicts tweets as suicidal and saves them in the database. The tweets predicted as non-suicidal are discarded. This sub-system runs in the background are parallelly to the other sub-system through multithreading.

### Frontend Application Sub-System

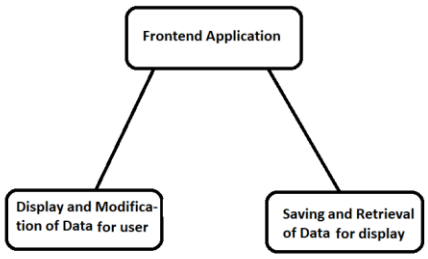


*Figure 5:1:4 Sub-system Architecture Design of Frontend Application*

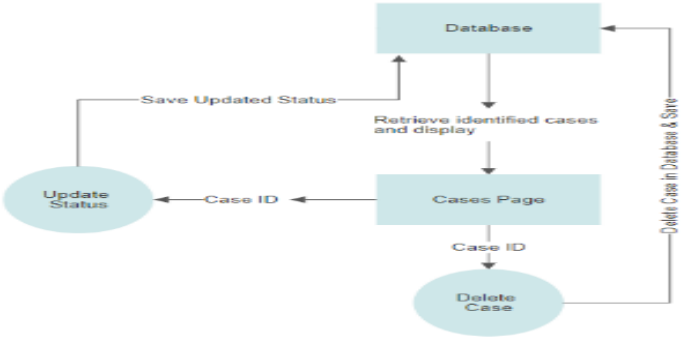


*Figure 5:1:5 Sub-system Data flow Diagram of Backend Application*

This sub-system displays the records saved in the database by the background application sub-system. It also allows the user to update and delete the records saved in the database. This sub-system runs in the foreground of the application for the user to interact with.

## 5.2 DETAILED SYSTEM DESING

This section provides the detailed system designing of the project.

### 5.2.1 Classification

All the designs are classified into different diagrams to be able to understood by the users

### 5.2.2 Definition

The process of establishing system aspects such as modules, architecture, components and their interfaces, and data for a system based on given requirements is known as systems design. It is the process of identifying, creating, and designing systems that meet a company's or organization's particular objectives and expectations.

### 5.2.3 Responsibilities

A systemic approach is essential for a well-functioning and cohesive system. To take into consideration all of the system's connected factors, a bottom-up or top-down strategy is necessary.

This System plays following important roles

- Save time
- Provides help to concerned authorities in identifying suicidal people
- Easy to use

### 5.2.4 Constraints

- Unified Modelling Language (UML): A graphical notation for describing software's structure and behavior.
- Flowchart: An algorithm's schematic or sequential depiction.

### 5.2.5 Composition

Methods of design:

- Architectural design: To explain the system's viewpoints, models, behavior, and structure.
- Logical design: To depict the system's data flow, inputs, and outputs.

### 5.2.6 Uses/Interactions

None

### 5.2.7 Resources

List of any and all the resources required is provided in the **Tables 3.2** and **3.3** of **Chapter 3**.

### 5.2.8 Processing

The SVM model is used for classification of tweets and identification of users through them.

### 5.2.9 Interface/Exports

The application created is named **The Psychiatrist**. The project's scope is as follows:

- The method must be effective and quick.
- Validation checks should be in place to prevent data entering errors.
- All of the data in the tables and forms should be presented in a professional manner.
- Records must be able to be added, updated and deleted.
- For the rising number of records, solved records must be deleted.
- The forms and tables should be simple to use.

### 5.2.10 Detailed Subsystem Design

The prototype is developed at this phase. Diagram of the system are provided in this section. The Use Case Diagram, Activity Diagram, data flow diagram (DFD) and Database Diagram are constructed. All of the diagrams serve as a guide for the system's flow. This application's database is made up of twitter users tweets information and the application users' information. These records are in a database will be used to create a prototype Basic functionality, such as displaying identified cases information, case status update and case delete functions.

## Diagrams

1. Use case Diagram

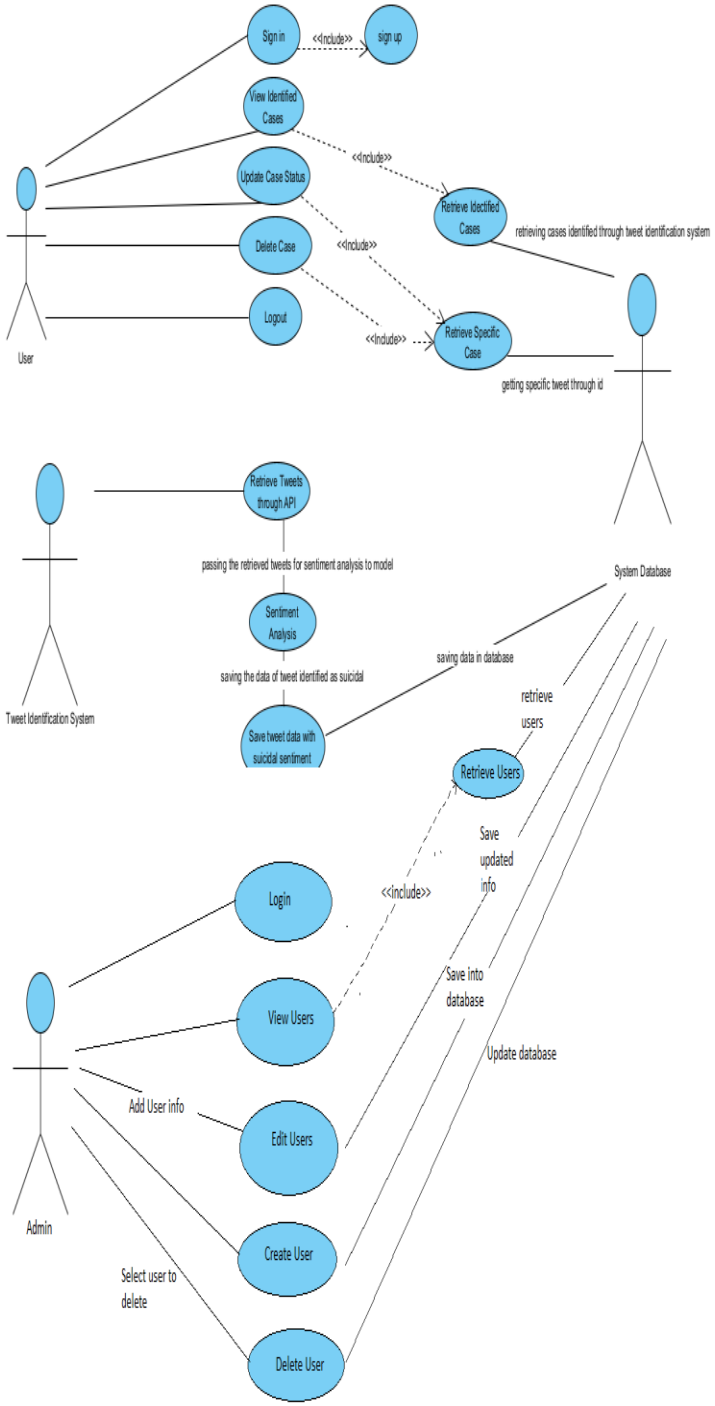


***Figure 5:2:1 Use case Diagram***

2. Activity Diagram

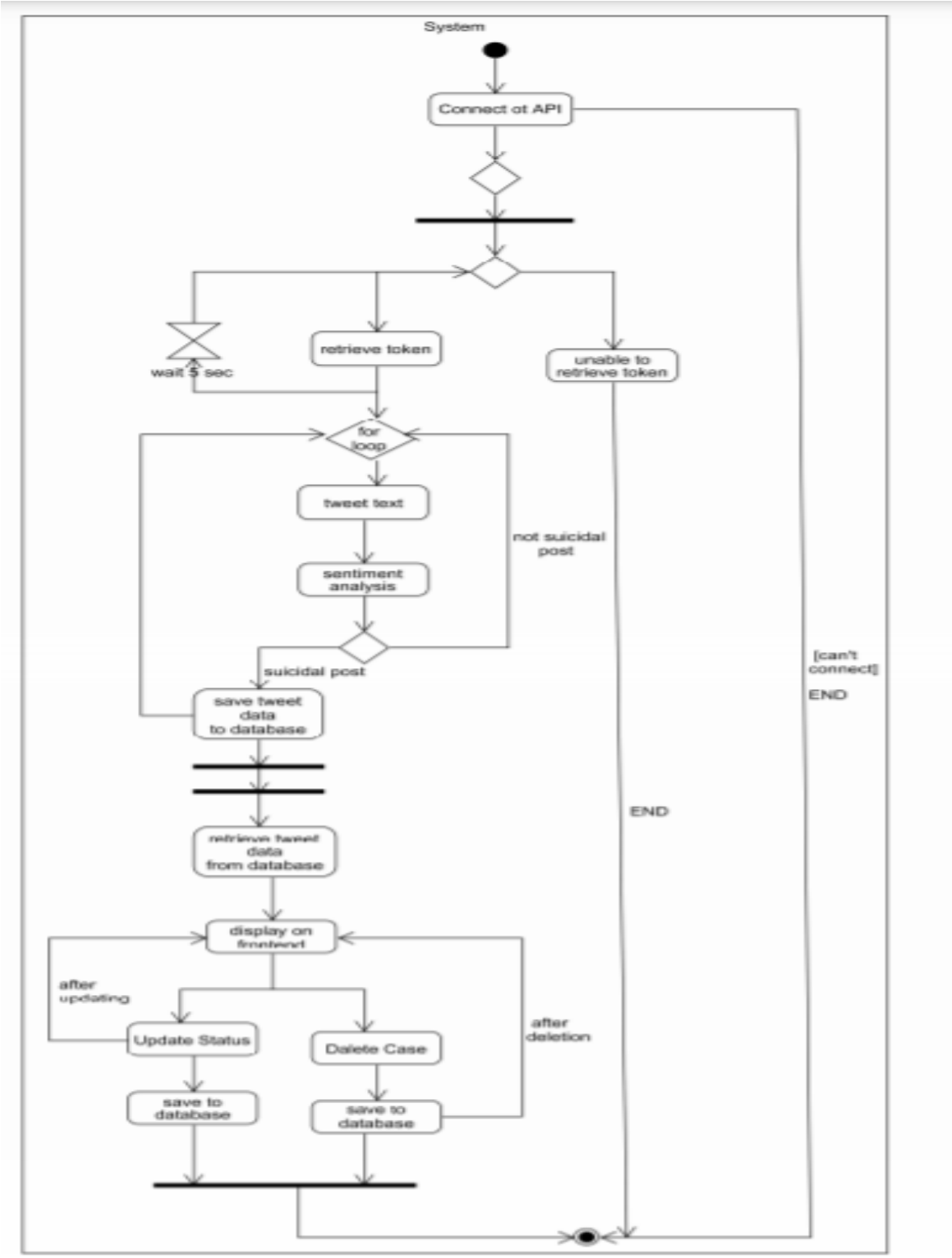


***Figure 5:2:2 Activity Diagram***

3. Data Flow Diagram

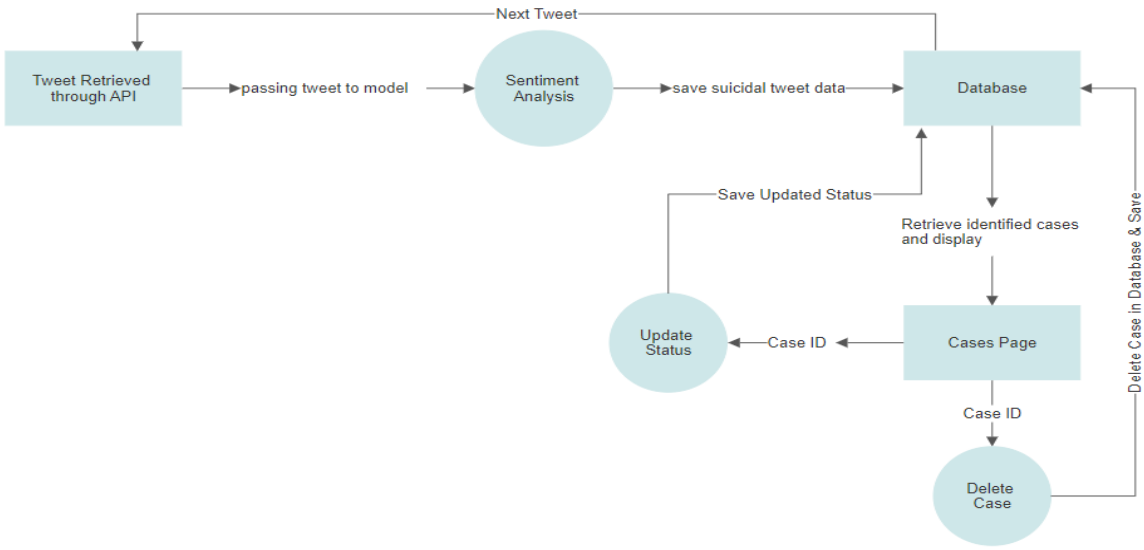


*Figure 5:2:3 Data Flow Diagram (DFD)*

4. Database Diagram

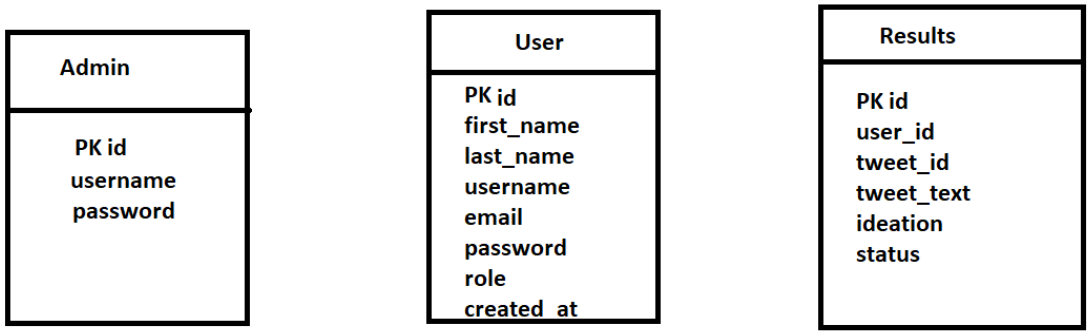


*Figure 5:2:4 Database Diagram*

# CHAPTRER-6

# Implementation and Testing

## 6.1 IMPLEMENTATION

### Twitter API:

Twitter API access is hard to get, it might take a while to get access to the Twitter's development API key. If you are not granted access to the API, you can't use this application.

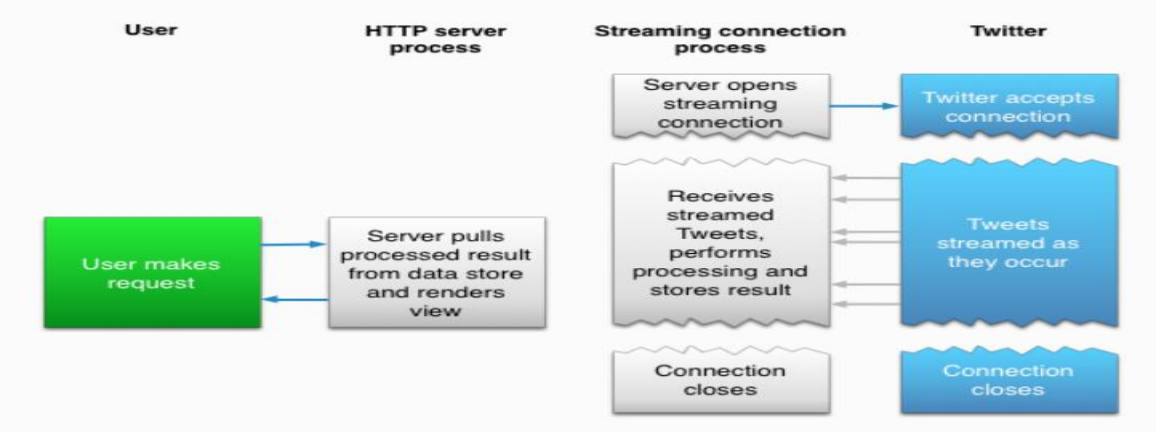


***Figure 6:0:1 How Twitter API works***

### Python:

Availability of Python language and related modules is also necessary as it will be used for sentiment analysis of the tweets.

### 6.1.1. Implementation

The GUI and the implementation of the application is provided below.

1. Install Git and open git bash terminal or Linux WSL 2 and enter the environment. For default python environment type

   **conda activate** or

   1. Create new environment

**python -m venv psychiatrist**

2. Activate the environment

   **source psychiatrist/Scripts/activate**

3. Install required packages

   **pip install -r project/requirements.txt**

In case of error while using Linux WSL2 run the following commands before installing requirements.txt file

1. **sudo apt-get update**
2. **sudo apt-get upgrade**
3. **sudo apt install python3-pip**
4. **sudo apt install python3.8-venv**


***Figure 6:2:1 Enter Development mode.***


***Figure 6:2:2 FLASK_APP.***


***Figure 6:2:3 Run Application.***

```
 * Serving Flask app "project" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Restarting with windowsapi reloader
2021-08-07 17:30:45,779 - helper - DEBUG - Config loaded successfully!
2021-08-07 17:30:47,349 - db - DEBUG - Database inititated successfully
 * Debugger is active!
2021-08-07 17:30:47,549 - helper - DEBUG - Config loaded successfully!
 * Debugger PIN: 912-730-195
2021-08-07 17:30:47,564 - werkzeug - INFO -  * Debugger PIN: 912-730-195
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
2021-08-07 17:30:47,629 - werkzeug - INFO -  * Running on http://127.0.0.1:5000/ (Press CTRL+C
to quit)
```

***Figure 6:2:4 APP URL.***


***Figure 6:2:5 Web-Frontend URL Paste.***

## 6.2. GUI

The GUI of each webpage of the application is given below. Image of each Webpage is provided on the next Page

### 6.2.1. User Panel:

Below is the User Panel of the Application.

1. HOME Page

   By default, at the start of the application this page will be displayed.

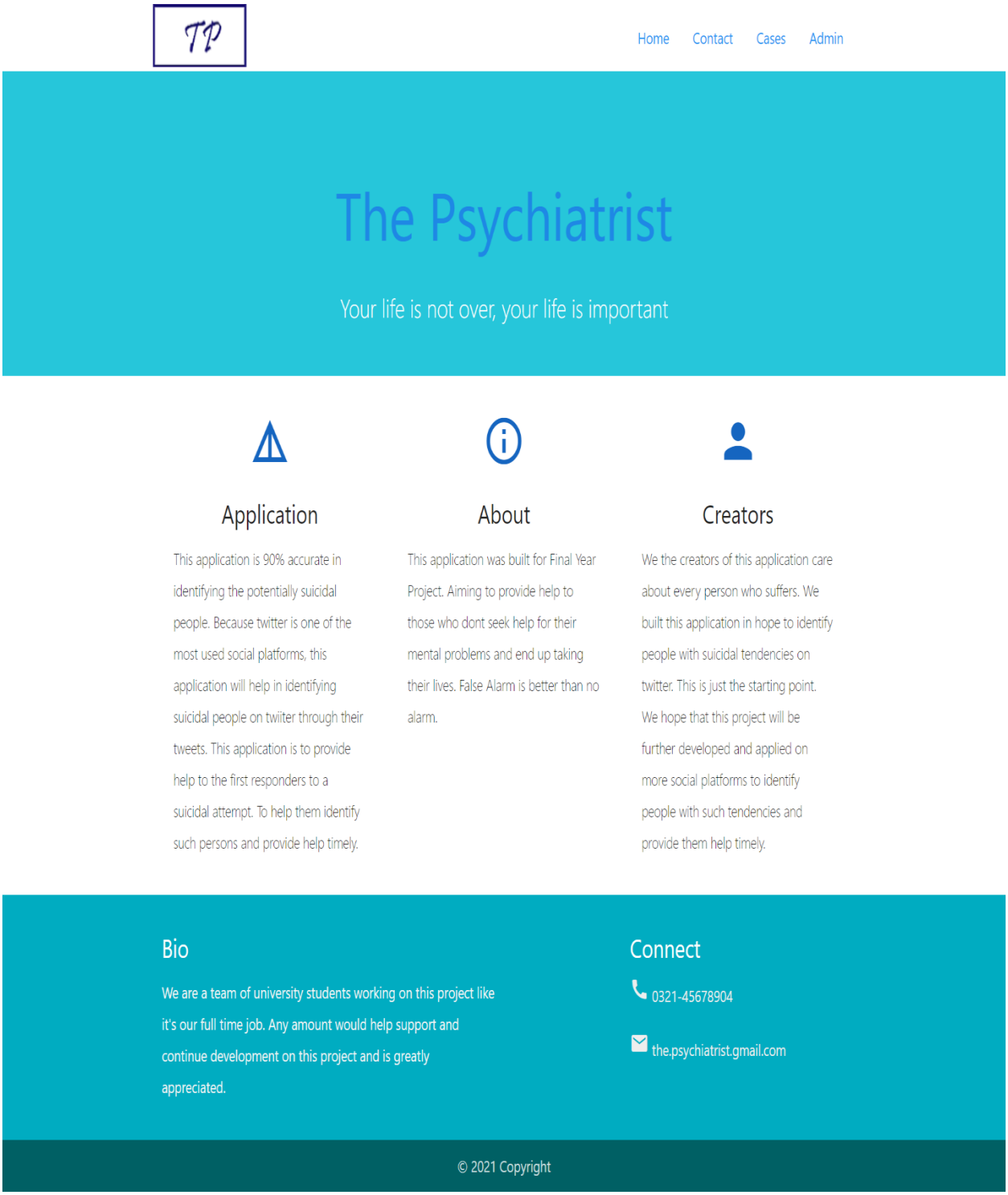


***Figure 6:2:6 HOME Page***

2. CONTACT PAGE

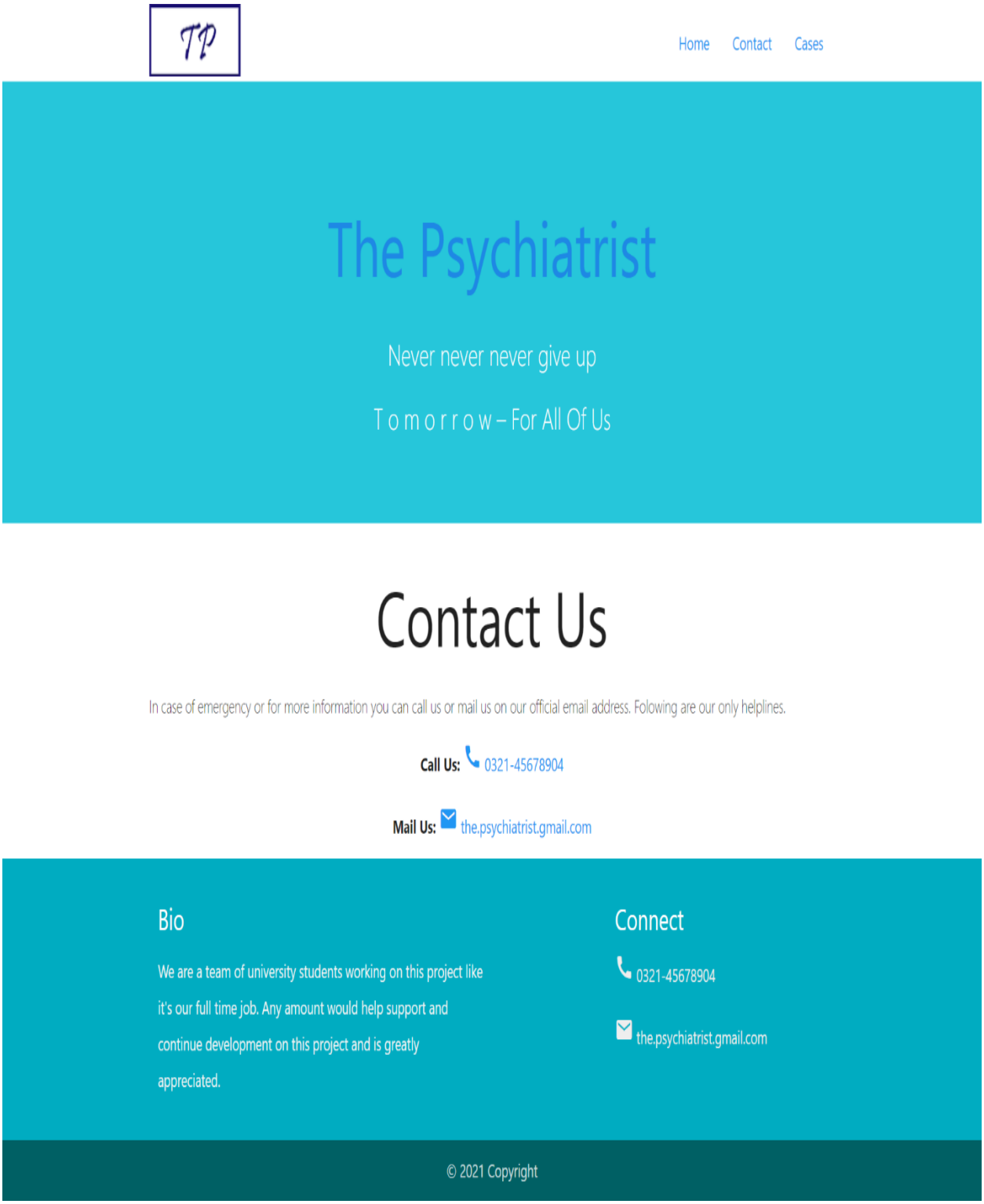


***Figure 6:2:7 CONTACT Page***

3. User LOGIN PAGE

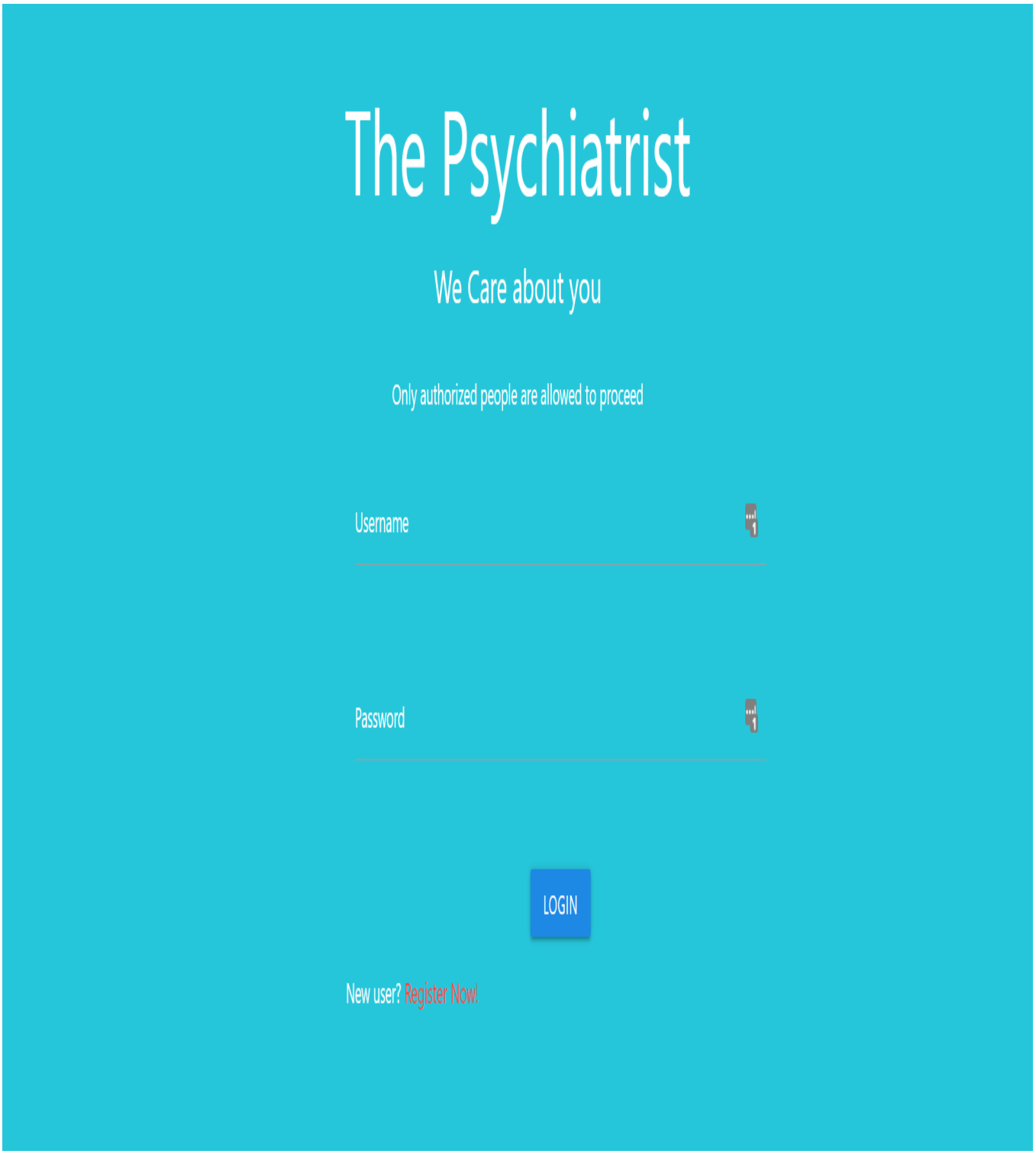


***Figure 6:2:8 On clicking the Cases tab at the top right corner, the login page is displayed.***

4. User SIGNUP Page

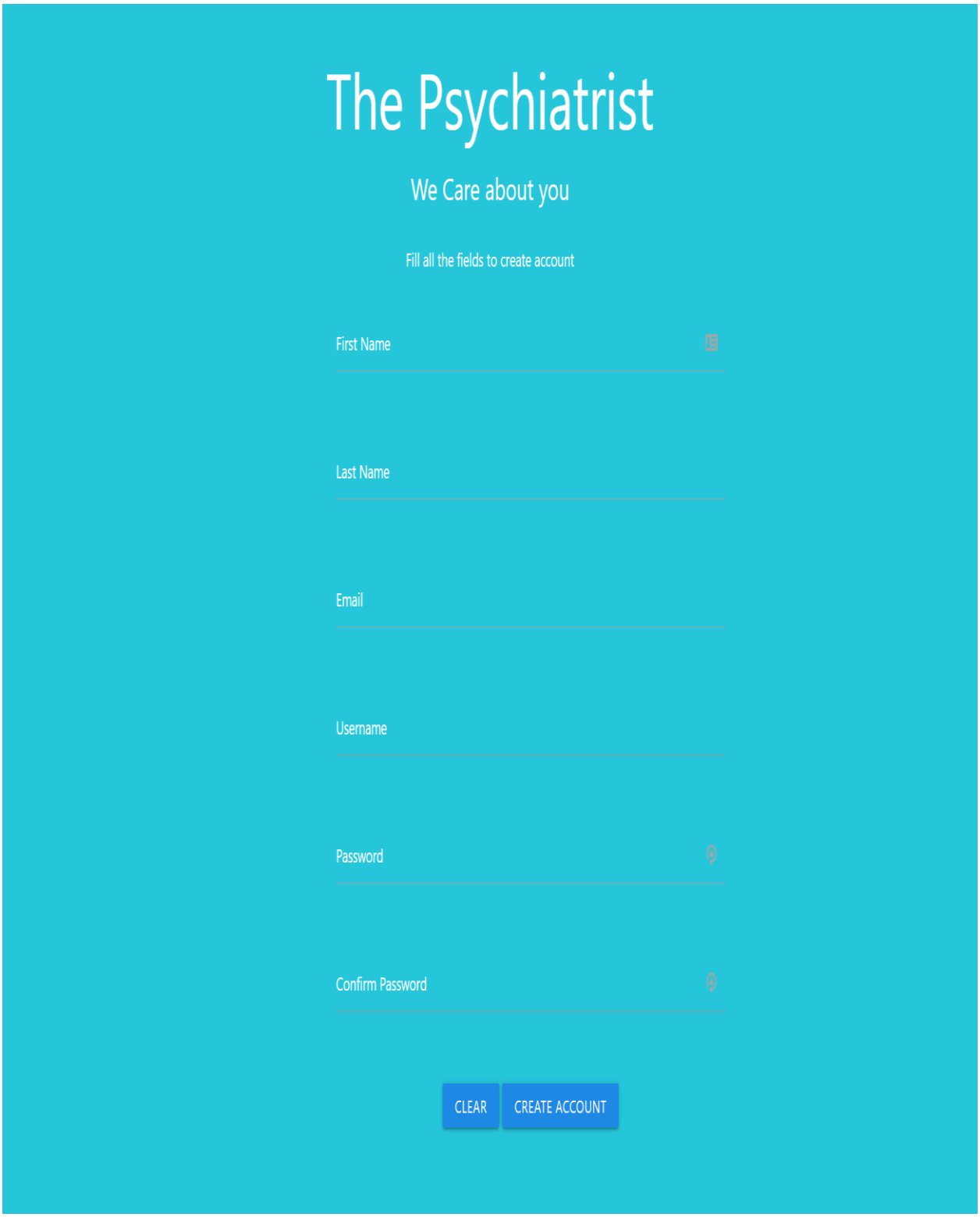


***Figure 6:2:9. On clicking Register Now at the top right corner, the signup page is displayed.***

5. CASES PAGE

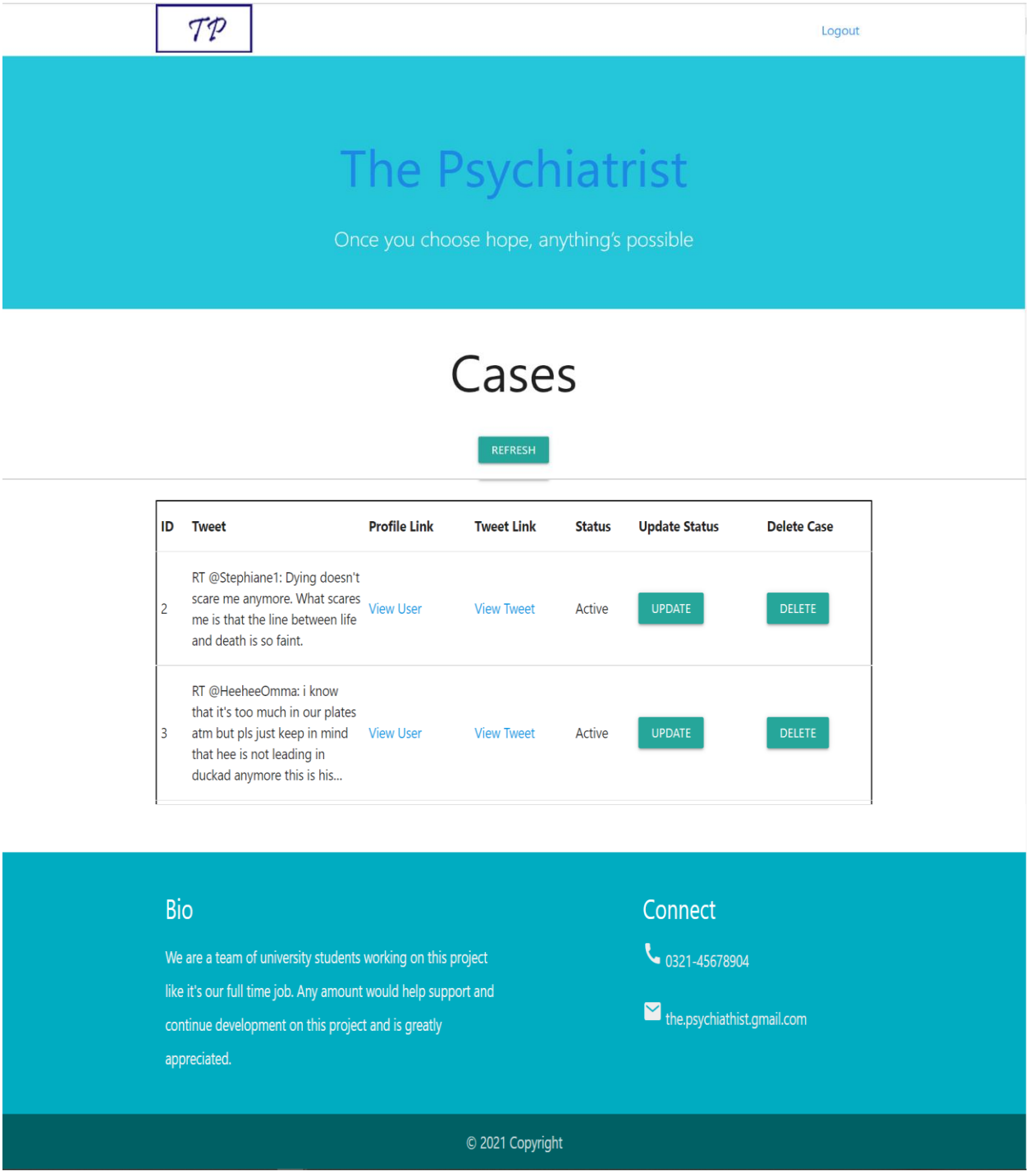


***Figure 6:2:10 After login, the authorized user can view identified cases on the SASES Page***

6. UPDATE STATUS PAGE

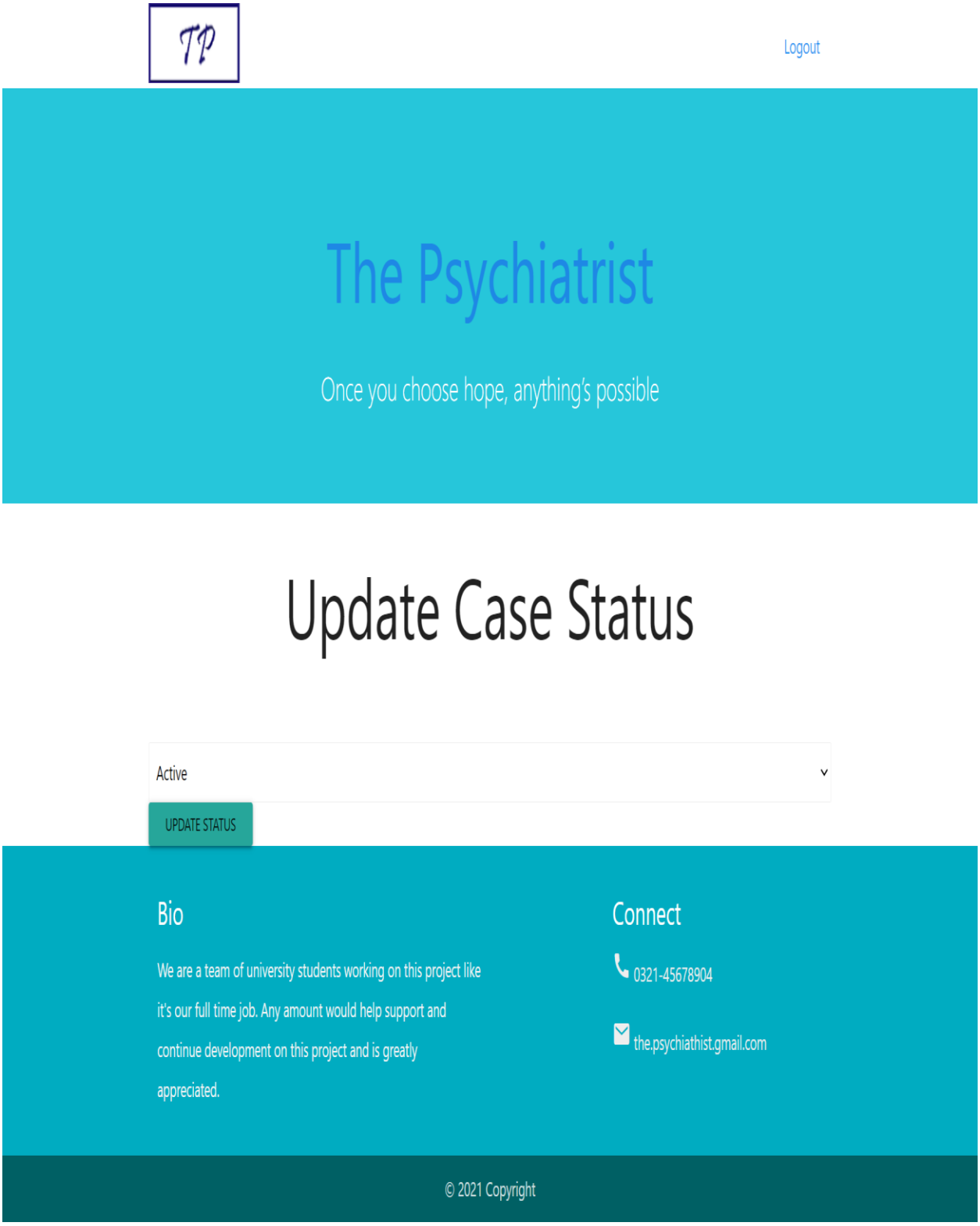


***Figure 6: 2:11 The case status can be changed as the case progresses between Active/In Progress/Closed.***

### 6.2.2. Admin Panel:

1. Admin Login

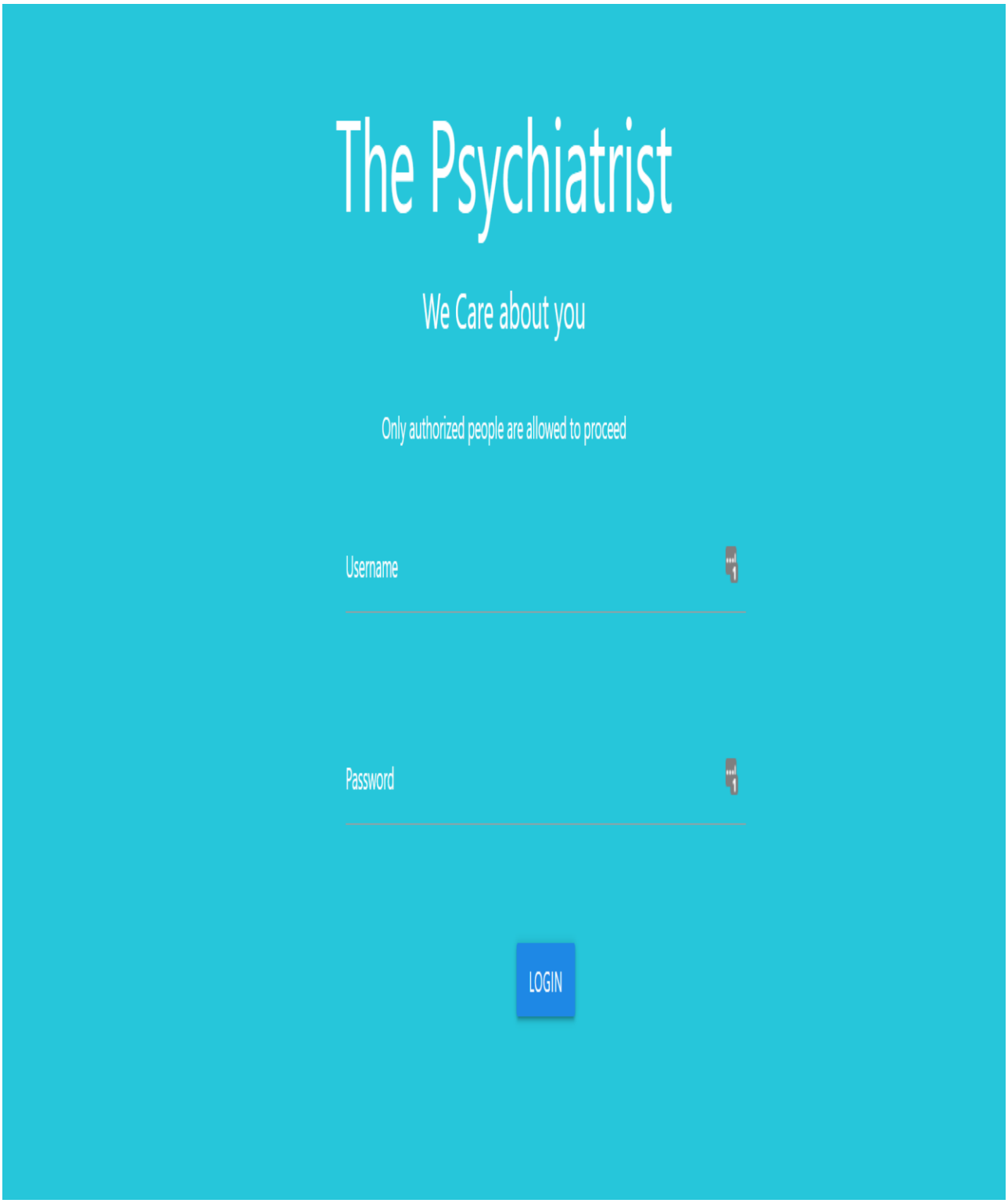


*Figure 6:2:12 Click on admin tab, admin login page opens*

2. Admin Dashboard

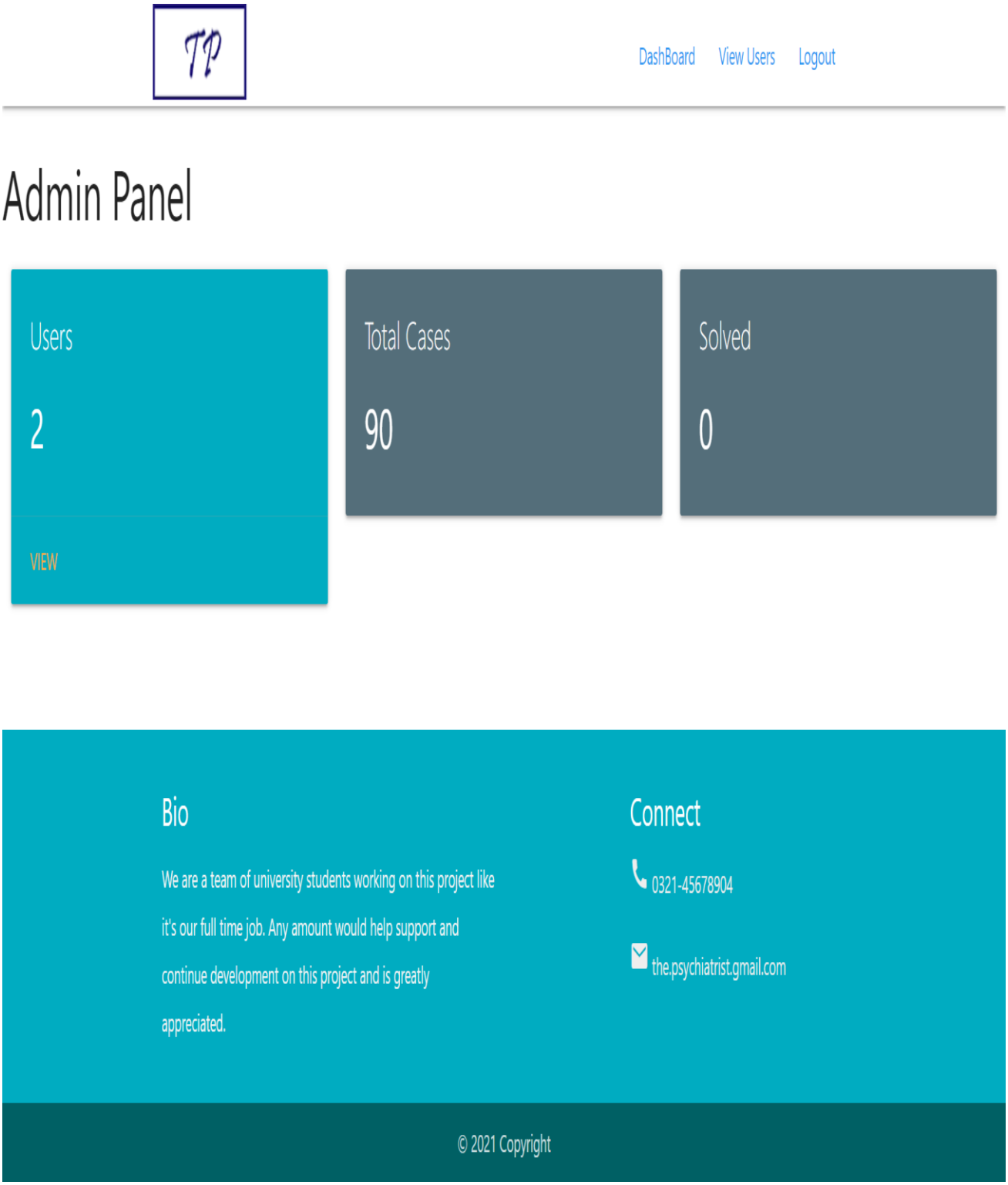


***Figure 6:2:13 After login, the admin dashboard opens***

3. User Management


***Figure 6:2:14 Click on View Users tab to view users and to perform CURD actions.***

4. Update User

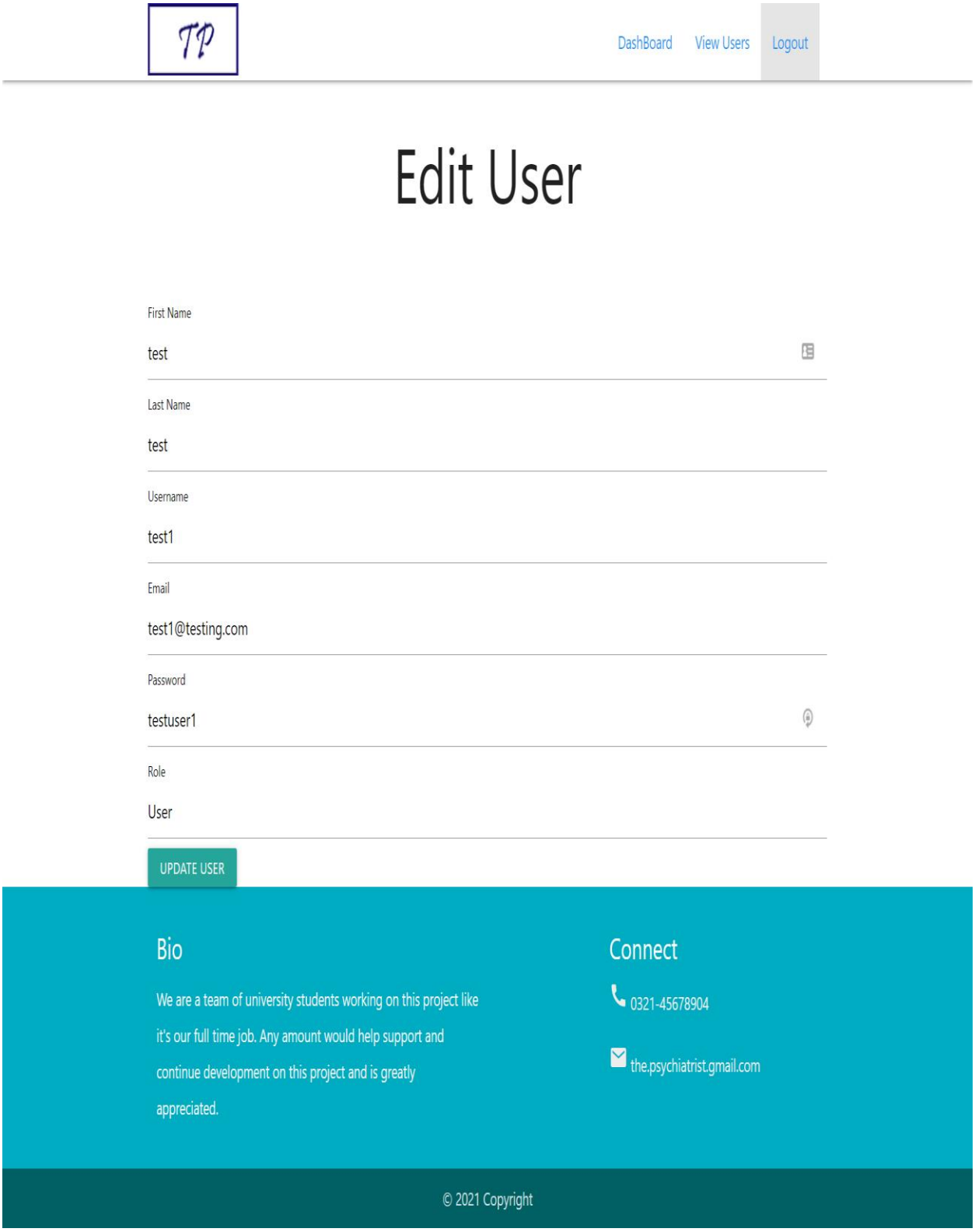


***Figure 6:2:15 Update/Edit User Information***

5. ADD/CREATE User

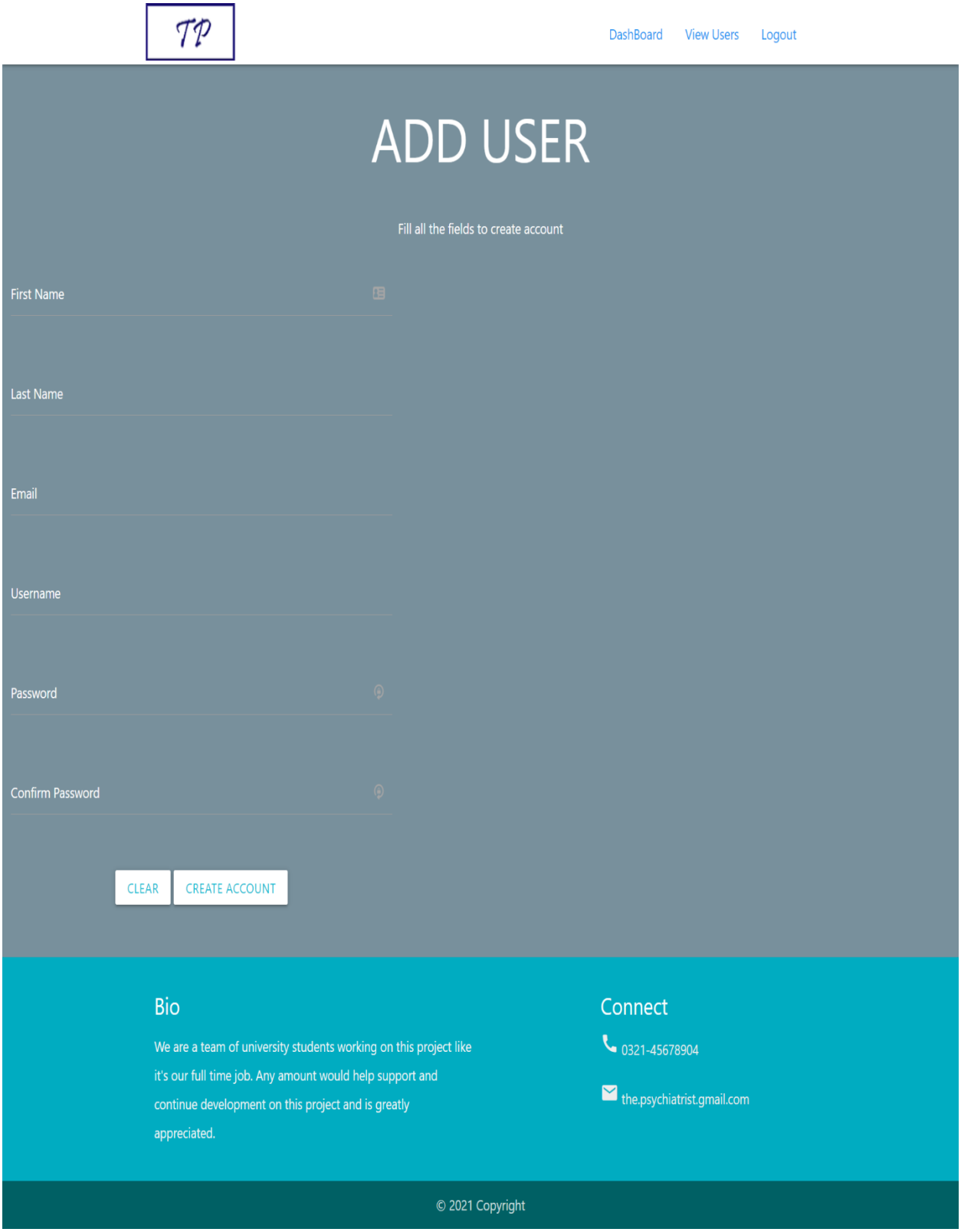


***Figure 6:2:16 Add/Create a new user***

Both Admin and User panels of the application provide DELETE functions.

# CHAPTER-7

# Results and Discussions

This chapter provides the overall results of the project. Furthermore, discussion points are also provided.

## 7.1. Results

Through this project, a method for identifying suicidal users on twitter is offered. To begin, real-time tweets retrieved from Twitter through its API are analyzed. Then a set of criteria was utilized to distinguish between suicidal and non-suicidal profiles. Through sentiment analysis of the tweet text user thoughts were examined, which can provide an insight into a suicidal profile's psychological state. This will enable concerned personals/officials to easily identify the twitter users with suicidal tendencies and contact required authorities to provide suitable medical attention and help the suffering people. The sentimental analysis results based on different queries including movie, politics, fashion, and fake news. The data based on the results we got form this step. If we run the program in different times we may get different results, small variance, based on the tweets we fetch. We run the program three times and these results are the average of the outputs. As it can be clearly seen in the table and diagram the percentage of the neutral tweets are significantly high. This is also important to mention that depends on the data of the experiment we may get different results as people’s opinion may change depends on the world circumstances for example fake news as it becomes the world of the year in 2017. For some queries, the neutral tweets are more than 60% which clearly shows the limitation of the current works. In this technical project, we discussed the importance of social network analysis and its applications in different areas. We focused on Twitter as and have implemented the python program to implement sentimental analysis. We showed the results on different daily topics. We realized that the neutral sentments are significantly high which shows there is a need to improve Twitter sentiment analysis.

Social networks is a rich platform to learn about people's opinion and sentiment regarding different topics as they can communicate and share their opinion actively on social medias including Facebook and Twitter. There are different opinionoriented information gathering systems which aim to extract people's opinion regarding different topics. The sentiment-aware systems these days have many applications from business to social siences.

## 7.2. Discussions

Nowadays, sentiment analysis or opinion mining could be a hot topic in machine learning. We have still so much to find regarding the feelings of corpus of texts terribly accurately thanks to the complexity within the English language. In this project we are tending to specialize in sentiments analysis. There is capability of labor within the range of sentiment analysis with slightly accepted background. For e.g. we tend to seen that clients as a rule utilize our site for explicit sorts of watchwords which can be partitioned into a couple of particular classes, to be specific: governmental issues/lawmakers, big names, items/brands, sports person, media and music. Subsequently we will attempt to perform separate feeling investigation on tweets that exclusively have a place with 1 of those categories (for example the training data wouldn't be general anyway explicit to 1 of those classifications) and analyze the outcomes we tend to get if we apply general sentiment analysis on that instead. Twitter's API is vastly helpful in data processing applications, and may offer large insights into the general public opinion if the Twitter API and large information analytics are a few things you've got more interest in. Twitter API can be used in most of the difficult sentiment gathering, involving people, trends, and social graphs that is very different for the human mind to get.

# CHAPTER-8

# Conclusion and Future Work

This chapter provides the overall conclusion of the project. Furthermore, pointers about future work that can be done on this project is also provided.

## 8.1. Conclusion

Identifying user profiles who are suicidal or show the signs of being suicidal was focused on in this project. A detection model was developed based on a collection of certain keywords. To identify Twitter accounts, linguistic and emotional features are used. Features were added in the application so that the identified user profile and tweet could be viewed on the twitter application by clicking on the provided links. Additional features were also added to change the identified case status as it progresses and delete function to delete solved cases to prevent database overload.

Machine Learning techniques (mostly deep learning) were applied to detect suicidal behaviour using classifiers. Furthermore, a Python-based system was employed to detect suicidal thoughts.

## 8.2. Future Work

We will to improve the outcomes of the suicidal profile identification in the future by defining the degree of suicidality more preciselyy, add more features like sending Direct Messages (DMs) to the identified users' followers about the user being identified as suicidal.

# References

[1] World Health Organization. *National Suicide Prevention Strategies: Progress, Examples and Indicators*; World Health Organization: Geneva, Switzerland, 2018.

[2] https://www.geeksforgeeks.org/software-engineering-architectural-design/

[3] Michael Mesfin Tadesse, “Detection of Suicide Ideation in Social Media Forums Using Deep Learning”, 2020.

# Phychiatrist

ORIGINALITY REPORT

| 9% | 6% | 0% | 6% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Higher Education Commission Pakistan<br>Student Paper | 2% |
| 2 | Submitted to Western International College (WINC London)<br>Student Paper | 1% |
| 3 | cs.gmu.edu<br>Internet Source | 1% |
| 4 | Submitted to Victorian Institute of Technology<br>Student Paper | 1% |
| 5 | Submitted to Istanbul Bilgi University<br>Student Paper | 1% |
| 6 | Submitted to Colorado Technical University Online<br>Student Paper | 1% |
| 7 | www.scribd.com<br>Internet Source | <1% |
| 8 | Submitted to National University of Singapore<br>Student Paper | <1% |

| | | |
|---|---|---|
| 9 | hdl.handle.net<br>Internet Source | <1 % |
| 10 | Submitted to Cranfield University<br>Student Paper | <1 % |
| 11 | www.snmmi.org<br>Internet Source | <1 % |
| 12 | media.proquest.com<br>Internet Source | <1 % |
| 13 | www.cisco.com<br>Internet Source | <1 % |
| 14 | www.mdpi.com<br>Internet Source | <1 % |
| 15 | www.coursehero.com<br>Internet Source | <1 % |
| 16 | docplayer.net<br>Internet Source | <1 % |
| 17 | www.91mobiles.com<br>Internet Source | <1 % |
| 18 | www.cse.chalmers.se<br>Internet Source | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |