A

Major Project

On

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING MACHINE LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

M.SHANMUKANJALI (197R1A05G9)

D.ARUN PRANEETH KUMAR (197R1A05D4)

P. VARSHEEK REDDY (197R1A05H6)

Under the Guidance of

Dr.T.S. MASTAN RAO

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2019-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING MACHINE LEARNING"** being submitted by **M.SHANMUKANJALI (197R1A05G9), D.ARUN PRANEETH KUMAR (197R1A05D4), P.VARSHEEK REDDY (197R1A05H6)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by them under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr.T.S. Mastan Rao
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on______________________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr.T.S. Mastan Rao,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to the Project Review Committee (PRC) **Dr. Punyaban Patel, Mrs. Shilpa, Dr. T. Subha Mastan Rao & J. Narasimha rao** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

M.SHANMUKANJALI	**(197R1A05G9)**
D.ARUN PRANEETH KUMAR	**(197R1A05D4)**
P. VARSHEEK REDDY	**(197R1A05H6)**

ABSTRACT

In the research of online social networks, detection of anonymous user behavior, detection of offensive data, etc. are traditional and important research works. This project is focused on the detection of offensive data, and bully statements in shared data of the social networks. For this system, using Machine Learning algorithms with Text Mining concepts predicted the offensive data to get more accurate results. This project proposed a system of “Cyber Bullying Detection (CBD) in Social Networking” for predicting bully data. This project is used two datasets, namely, ‘Hate Speech and Offensive Language Dataset’ and ‘Harassment-Corpus Dataset’. This project used three Machine Learning classifiers such as Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), and Neural Network (NN) Algorithms and calculated performance results for comparing the performance for both datasets. To demonstrate this system designed and developed a Python-based Django web application and showed the results.

LIST OF FIGURES/TABLES

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

Most chat applications in the internet like WhatsApp, Messenger or other social networking apps offering chat communication tools for text messages, media data sharing, web data sharing etc. In current trend many social networking sites created and providing services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provide major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produces per day. According to research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc [1]. But in the current social sites not focus on services like tracking the user behavior of anonymous behavior. In current system, social network sites need to focus the user microblogs and need to capture the user behavior whether he/she anonymous user or not. Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time, applying those concepts in social networks is less practical. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information. In social networks users may share their messages by using the chat applications. For every social networking sites has their own chat applications, for this facebook is main example. And another way is sharing the multimedia data like images or videos. For this Facebook and Instagram best examples. For communication between users chat applications will most useful for share their information, thoughts, views etc. But in the same way it may also cause the security loophole of user's security which is cyber bullying. Such text based content may security threat to the users because of the people can share cyber bullying words to the users with their fake accounts. [2], [3]. Based on these disadvantages detection malicious users is active topic in the study of social media.

1.1 RELATED WORK

In the current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provides major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produced per day. According to a research survey, more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc [1]. But the current social sites do not focus on services like tracking the user behavior or anonymous behavior. In the current system, social network sites need to focus on the user microblogs and need to capture the user behavior whether he/she is an anonymous user or not.

Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time, applying those concepts in social networks is less practical. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

In the current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous user accounts are being created very rapidly. We need to focus on tracking the anonymous users. In our proposed system we are implementing a web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she does with others. By taking advantage of Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

In current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provides major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produced per day. According to a research survey many more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc. But the current social sites do not focus on services like tracking the user behavior or anonymous behavior. In the current system, social network sites need to focus on the user microblogs and need to capture the user behavior whether his/she is an anonymous user or not. Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time, applying those concepts in social network is less practical.

2.2 PROPOSED SYSTEM:

The proposed system 'Cyberbullying Detection (CBD) in Social Networking' is to identify the bully or offensive statements using a classification model. Based on this requirement, the proposed architecture is designed based on two flows, namely, classification analysis and user side prediction. Based on the project requirement, this architecture is designed which is represented in figure 1. This section described the workflow of the architecture and project main modules. The main purpose of the CBD system is to identify the bully or offensive statements using Machine Learning algorithms. To achieve this requirement, we need to implement classification analysis. The classification analysis is the process of conducting training and testing processes for calculating the performance measures between various Machine Learning algorithms. Based on these requirements, the proposed architecture is designed with two flows, namely, classification analysis which is taken care of by the admin and user side prediction. In the architecture these two flows are represented in two different color formats.

2.3 DISADVANTAGES OF EXISTING SYSTEM:

Based on existing surveys, attackers easily morph the data.

- Very less practical, we can’t find the attacker using small tiny blogs
- Based on the network based models we should effort heavy data for detecting

2.4 ADVANTAGES OF PROPOSED SYSTEM:

We are depending on the chat history instead of the profile attributes or any other media attributes, so we can conclude with a minimum amount of the computation.

- Due to machine learning classifications we can get accurate results

- Cyberbullying detection process is automatic and time taken for detection is less and it works on the live environment.

- The latest machine learning models are used for training models that are accurate.

2.5 HARDWARE & SOFTWARE REQUIREMENTS:

2.5.1 HARDWARE REQUIREMENTS:

RAM	**4 GB Minimum**
Processor	**i3 Minimum**
Hard disk	**500 GB**

2.5.2 SOFTWARE REQUIREMENTS:

Technology	**Python 3.6**
Operating System	**Windows Family**
IDE	**VS Code**
Technology	**Python, Django**
Database Server	**MySQL**
Front Design Technology	**HTML, CSS, JS**

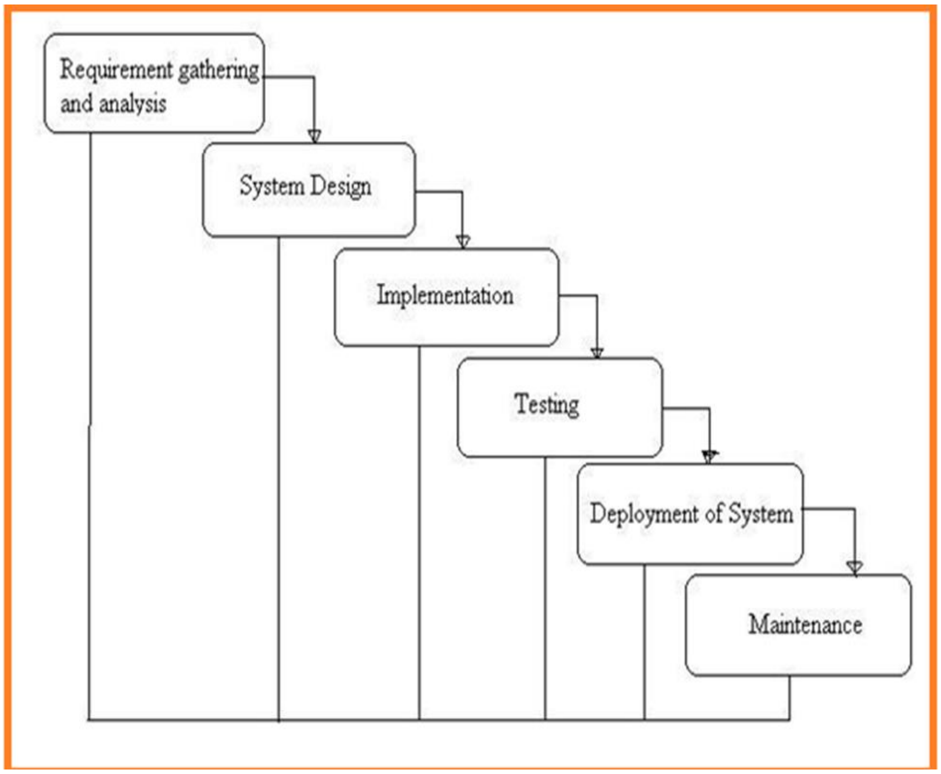


Fig:-1 Project SDLC

- Project Requisites Accumulating and Analysis
- Application System Design
- Practical Implementation
- Manual Testing of My Application
- Application Deployment of System
- Maintenance of the Project

2.6 REQUISITES ACCUMULATING AND ANALYSIS

It’s the first and foremost stage of the any project as our is a an academic leave for requisites amassing we followed of IEEE Journals and Amassed so many IEEE Relegated papers and final culled a Paper designated “Individual web revisitation by setting and substance importance inputand for analysis stage we took referees from the paper and did literature survey of some papers and amassed all the Requisites of the project in this stage.

2.7 SYSTEM DESIGN

In System Design has divided into three types like GUI Designing, UML Designing with avails in development of project in facile way with different actor and its utilizer case by utilizer case diagram, flow of the project utilizing sequence, Class diagram gives information about different class in the project with methods that have to be utilized in the project if comes to our project our UML Will utilizable in this way The third and post import for the project in system design is Data base design where we endeavor to design data base predicated on the number of modules in our project.

2.8 IMPLEMENTATION

The Implementation is Phase where we endeavor to give the practical output of the work done in designing stage and most of Coding in Business logic lay coms into action in this stage its main and crucial part of the project.

TESTING:

Unit Testing

It is done by the developer itself in every stage of the project and fine-tuning the bug and module predicated additionally done by the developer only here we are going to solve all the runtime errors.

Manual Testing

As our Project is academic Leave we can do any automatic testing so we follow manual testing by endeavor and error methods.

Deployment of System

Once the project is total yare we will come to deployment of client system in genuinely world as its academic leave we did deployment i our college lab only with all need Software’s with having Windows OS

Maintenance

The Maintenance of our Project is one time process onl

3. ARCHITECTURE

3. ARCHITECTURE

In this architecture I describe the flow of the process of the project. This architecture will describe the online social user process of the chat procedure and detection of anonymous users, in this flow there is a manual procedure and another one is system procedure in the detection process. Let's describe the architecture components.

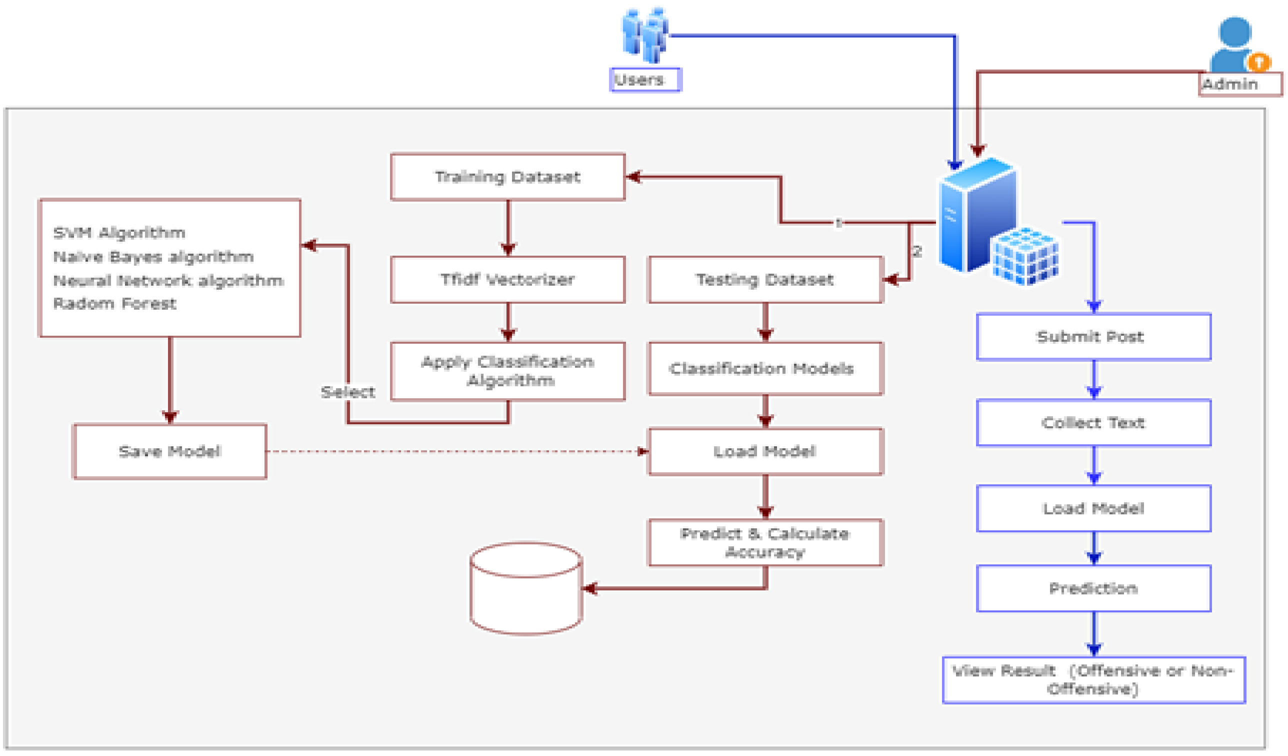


Fig:-1 CBD Architecture

The proposed system 'Cyberbullying Detection (CBD) in Social Networking' is to identify the bully or offensive statements using a classification model. Based on this requirement, the proposed architecture is designed based on two flows, namely, classification analysis and user side prediction. Based on the project requirement, this architecture is designed which is represented in figure 1. This section described the workflow of the architecture and project main modules. The main purpose of the CBD system is to identify the bully or offensive statements using Machine Learning algorithms. To achieve this requirement, need to implement classification analysis.

MODULES:

Admin:

Admin is a main user of our application, admin will process main functionalities of the system. Admin can see the lists of users are available in the system. Admin can process the detection of anomaly users with Naïve Bayes algorithm. Admin will send a warning mail to the user. After that user if continue the same manner then admin will block the account of the anomaly user.

User:

In our system user is an end user of our application. We build a social application for users. They can make friends and share data among them. User also chat with his/her friends. When a user sends any bullying words to the others application will detect the user and send the warning mail to the user. User will get the notification of warning through mail. After getting the warning alert if user will continue the same manner then user will blocked in the system.

SYSTEM DESIGN:

UML DIAGRAMS:

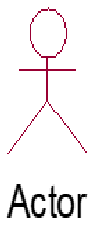
The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

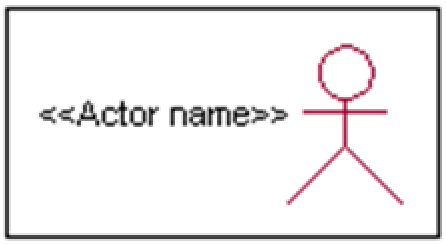
Global Use Case Diagrams:

Identification of actors:

Actor**:** Actor represents the role a user plays with respect to the system. An actor interacts with, but has no control over the use cases.

Graphical representation:





An actor is someone or something that:

Interacts with or uses the system.

- Provides input to and receives information from the system.
- Is external to the system and has no control over the use cases.

Actors are discovered by examining:

- Who directly uses the system?
- Who is responsible for maintaining the system?
- External hardware used by the system.
- Other systems that need to interact with the system.

Questions to identify actors:

- Who is using the system? Or, who is affected by the system? Or, which groups need help from the system to perform a task?
- Who affects the system? Or, which user groups are needed by the system to perform its functions? These functions can be both main functions and secondary functions such as administration.
- Which external hardware or systems (if any) use the system to perform tasks?
- What problems does this application solve (that is, for whom)?
- And, finally, how do users use the system (use case)? What are they doing with the system?

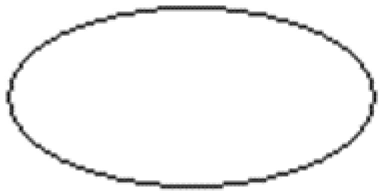
The actors identified in this system are:

a. **System Administrator**
b. **Customer**
c. **Customer Care**

Identification of use cases:

Use Case: A use case can be described as a specific way of using the system from a user's (actor's) perspective.

Graphical representation:



A more detailed description might characterize a use case as:

- Pattern of behavior the system exhibits
- A sequence of related transactions performed by an actor and the system
- Delivering something of value to the actor

Use cases provide a means to:

- capture system requirements
- communicate with the end users and domain experts
- test the system

Use cases are best discovered by examining the actors and defining what the actor will be able to do with the system.

Guidelines for identifying use cases:

- For each actor, find the tasks and functions that the actor should be able to perform or that the system needs the actor to perform. The use case should represent a course of events that leads to clear goal
- Name the use cases.
- Describe the use cases briefly by applying terms with which the user is familiar.

This makes the description less ambiguous

Questions to identify use cases:

- What are the tasks of each actor?
- Will any actor create, store, change, remove or read information in the system?
- What use case will store, change, remove or read this information?
- Will any actor need to inform the system about sudden external changes?
- Does any actor need to inform about certain occurrences in the system?

Flow of Events

A flow of events is a sequence of transactions (or events) performed by the system. They typically contain very detailed information, written in terms of what the system should do, not how the system accomplishes the task. Flow of events are created as separate files or documents in your favorite text editor and then attached or linked to a use case using the Files tab of a model element.

A flow of events should include:

- When and how the use case starts and ends
- Use case/actor interactions
- Data needed by the use case
- Normal sequence of events for the use case
- Alternate or exceptional flows

Construction of Use Case diagrams:

Use-case diagrams graphically depict system behavior (use cases). These diagrams present a high level view of how the system is used as viewed from an outsider's (actor's) perspective. A use-case diagram may depict all or some of the use cases of a system.

A use-case diagram can contain:

- actors ("things" outside the system)
- use cases (system boundaries identifying what the system should do)
- Interactions or relationships between actors and use cases in the system including the associations, dependencies, and generalizations.

Relationships in use cases:

1. Communication:

The communication relationship of an actor in a use case is shown by connecting the actor symbol to the use case symbol with a solid path. The actor is said to communicate with the use case.

2. Uses:

A Uses relationship between the use cases is shown by the generalization arrow from the use case.

3. Extends:

The extended relationship is used when we have one use case that is similar to another use case but does a bit more. In essence it is like a subclass.

SEQUENCE DIAGRAMS

A sequence diagram is a graphical view of a scenario that shows object interaction in a time-based sequence what happens first, what happens next. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces.
There are two main differences between sequence and collaboration diagrams: sequence diagrams show time-based object interaction while collaboration diagrams show how objects associate with each other.

Object:

An object has state, behavior, and identity. The structure and behavior of similar objects are defined in their common class. Each object in a diagram indicates some instance of a class. An object that is not named is referred to as a class instance.

The object icon is similar to a class icon except that the name is underlined:
An object's concurrency is defined by the concurrency of its class.

Message:

A message is the communication carried between two objects that trigger an event. A message carries information from the source focus of control to the destination focus of control. The synchronization of a message can be modified through the message specification. Synchronization means a message where the sending object pauses to wait for results.

Link:

A link should exist between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another. The link is depicted as a straight line between objects or objects and class instances in a collaboration diagram. If an object links to itself, use the loop version of the icon.

CLASS DIAGRAM:

Identification of analysis classes:

A class is a set of objects that share a common structure and common behavior (the same attributes, operations, relationships and semantics). A class is an abstraction of real-world items.

There are 4 approaches for identifying classes:

a. Noun phrase approach.
b. Common class pattern approach.
c. Use case Driven Sequence or Collaboration approach.
d. Classes , Responsibilities and collaborators Approach.

1. Noun Phrase Approach:

The guidelines for identifying the classes:

- Look for nouns and noun phrases in the usecases.
- Some classes are implicit or taken from general knowledge.
- All classes must make sense in the application domain; Avoid computer implementation classes defer them to the design stage.
- Carefully choose and define the class names After identifying the classes we have to eliminate the following types of classes: Adjective classes.

2. Common class pattern approach:

The following are the patterns for finding the candidate classes:

- Concept class.
- Events class.
- Organization class
- Peoples class
- Places class

3. Use case driven approach:

We have to draw the sequence diagram or collaboration diagram. If there is a need for some classes to represent some functionality then add new classes which perform those functionalities.

4. CRC approach:

The process consists of the following steps:

- Identify classes' responsibilities (and identify the classes)
- Assign the responsibilities
- Identify the collaborators.

Identification of responsibilities of each class:

The questions that should be answered to identify the attributes and methods of a class respectively are:

a. What information about an object should we keep track of?
b. What services must a class provide?

Identification of relationships among the classes:

Three types of relationships among the objects are:

Association: How objects are associated?

Super-sub structure: How are objects organized into super classes and sub classes?

Aggregation: What is the composition of the complex classes?

Association:

The questions that will help us to identify the associations are:

a. Is the class capable of fulfilling the required task by itself?
b. If not, what does it need?
c. From what other classes can it acquire what it needs?

Guidelines for identifying the tentative associations:

- A dependency between two or more classes may be an association. Association often corresponds to a verb or prepositional phrase.
- A reference from one class to another is an association. Some associations are implicit or taken from general knowledge.

Some common association patterns are:

Location association like part of, next to, contained in…..

Communication association like talk to, order to ……

We have to eliminate the unnecessary associations like implementation associations, ternary or n-ary associations and derived associations.

Super-sub class relationships:

Super-sub class hierarchy is a relationship between classes where one class is the parent class of another class (derived class).This is based on inheritance.

Guidelines for identifying the super-sub relationship, a generalization are

1. Top-down:

Look for noun phrases composed of various adjectives in a class name. Avoid excessive refinement. Specialize only when the sub classes have significant behavior.

2.Bottom-up:

Look for classes with similar attributes or methods. Group them by moving the common attributes and methods to an abstract class. You may have to alter the definitions a bit.

3.Reusability:

Move the attributes and methods as high as possible in the hierarchy.

4. Multiple inheritances:

Avoid excessive use of multiple inheritances. One way of getting benefits of multiple inheritances is to inherit from the most appropriate class and add an object of another class as an attribute.

Aggregation or a-part-of relationship:

It represents the situation where a class consists of several component classes. A class that is composed of other classes doesn’t behave like its parts. It behaves very difficultly. The major properties of this relationship are transitivity and anti symmetry.

The questions whose answers will determine the distinction between the part and whole relationships are:

- Does the part class belong to the problem domain?
- Is the part class within the system’s responsibilities?
- Does the part class capture more than a single value?(If not then simply include it as an attribute of the whole class)
- Does it provide a useful abstraction in dealing with the problem domain?

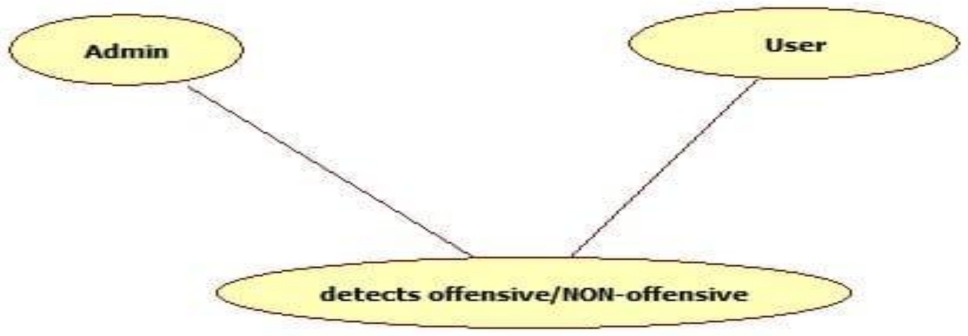
There are three types of aggregation relationships. They are:

Assembly:
It is constructed from its parts and an assembly-part situation physically exists.

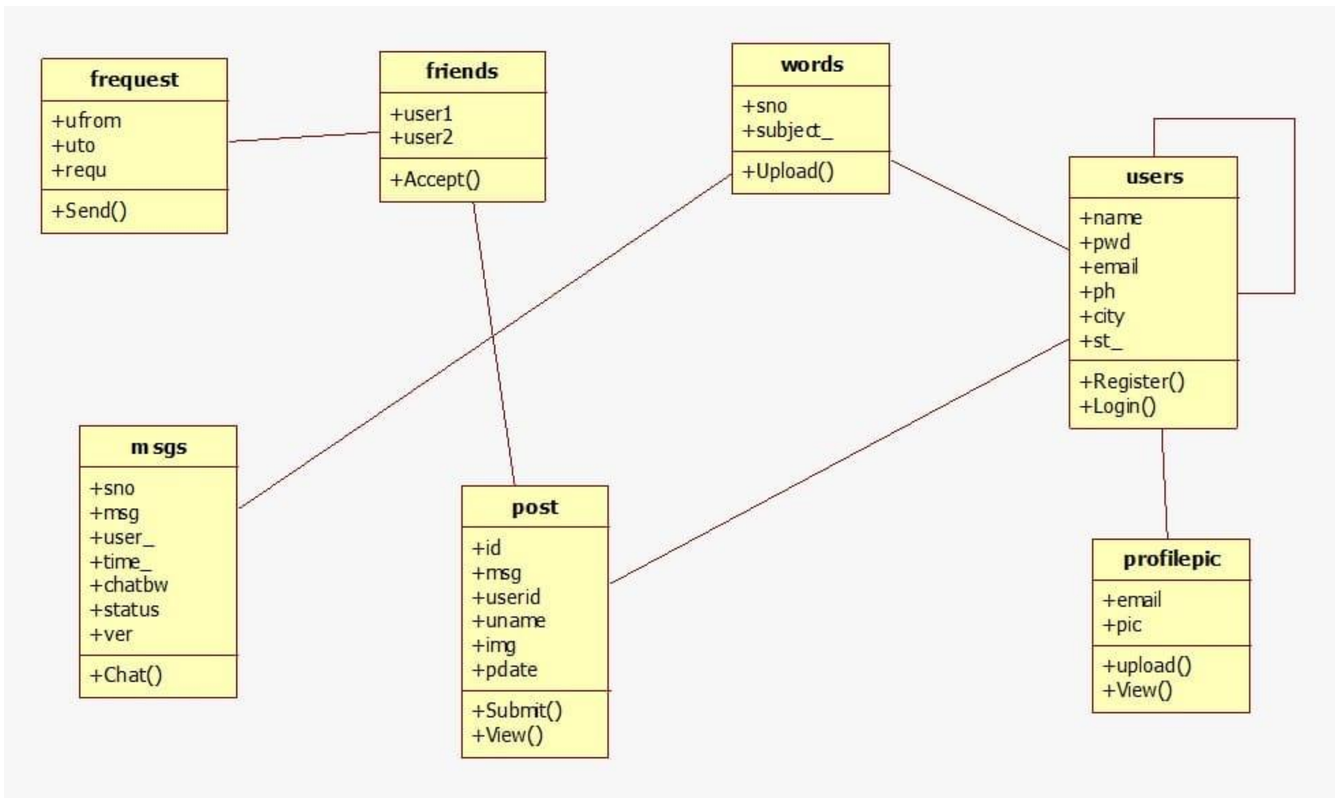
Container:
A physical whole encompasses but is not constructed from physical parts.

Collection member:
A conceptual whole encompasses parts that may be physical or conceptual. The container and collection are represented by hollow diamonds but composition is represented by solid diamonds.

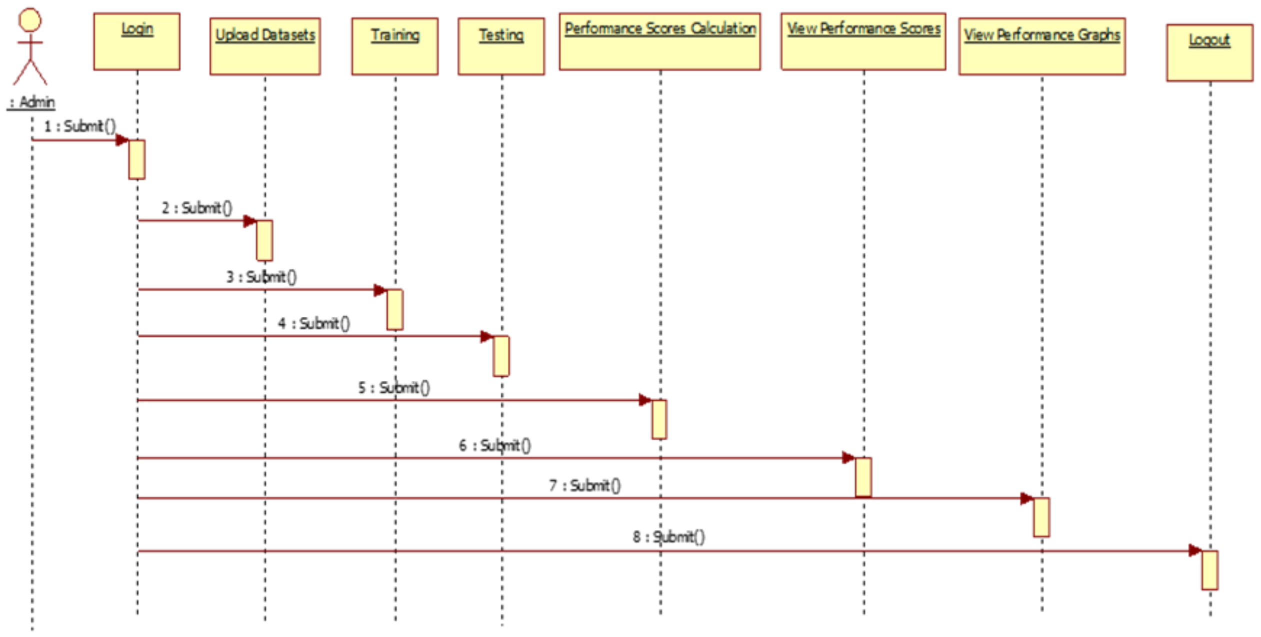
3.2 USE CASE DIAGRAM



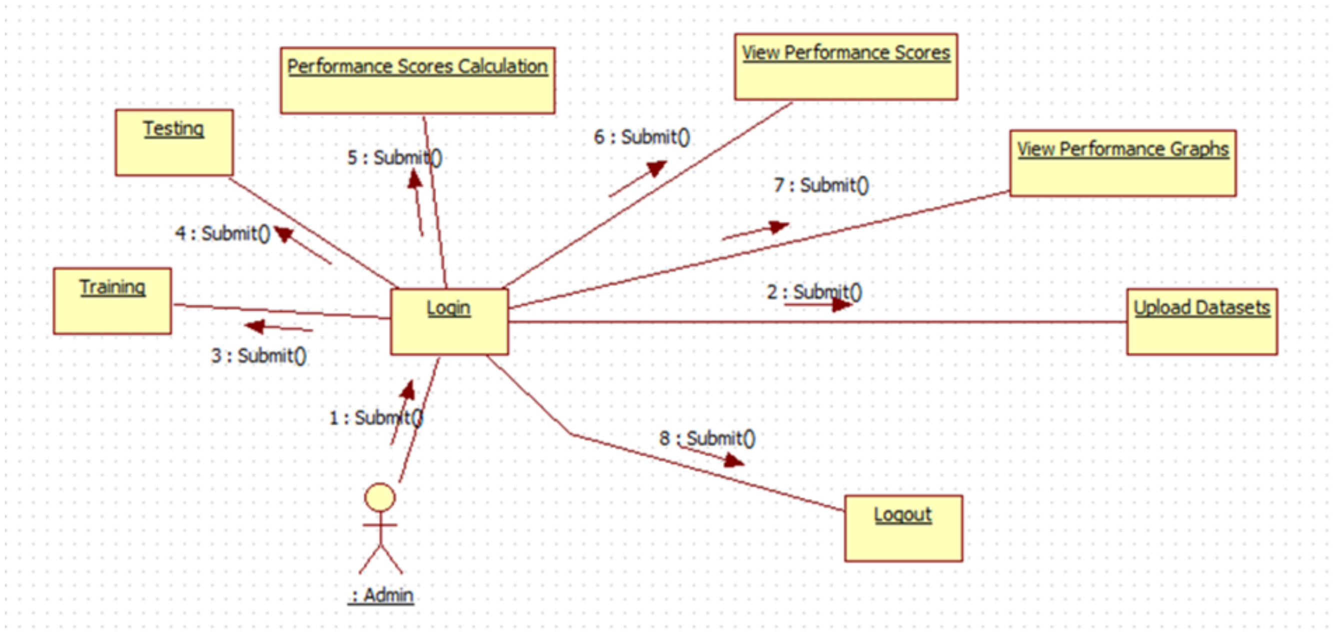
3.3 CLASS DIAGRAM



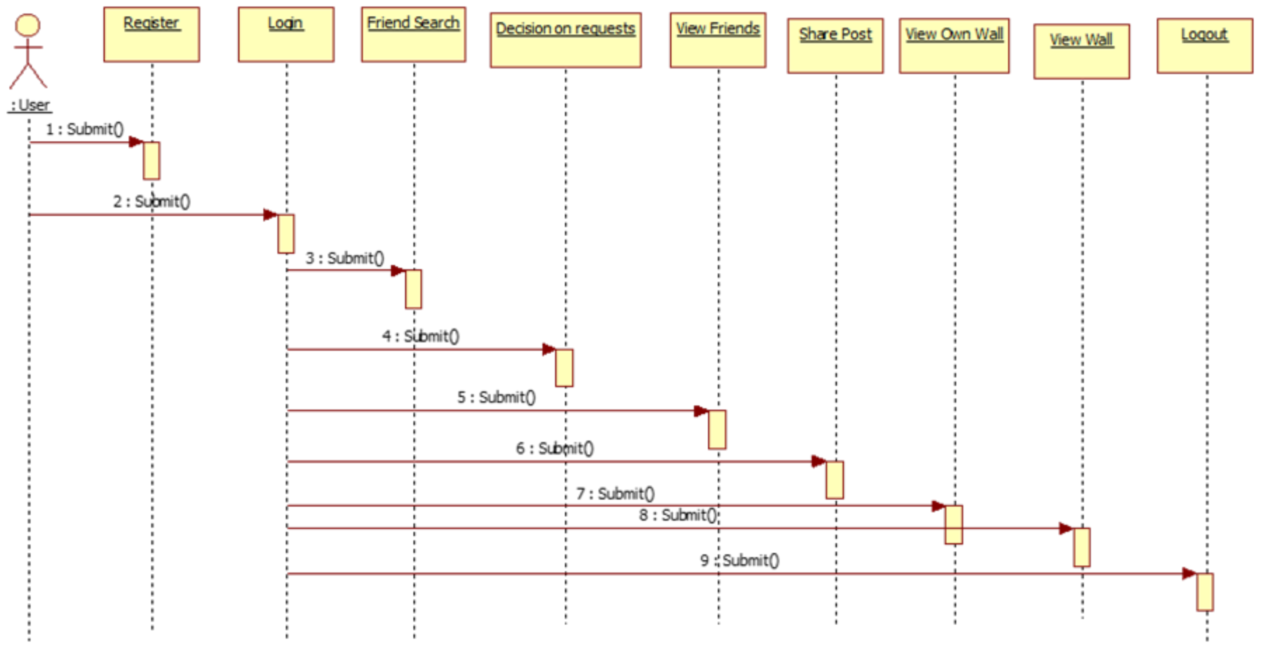
3.4 ADMIN SEQUENCES



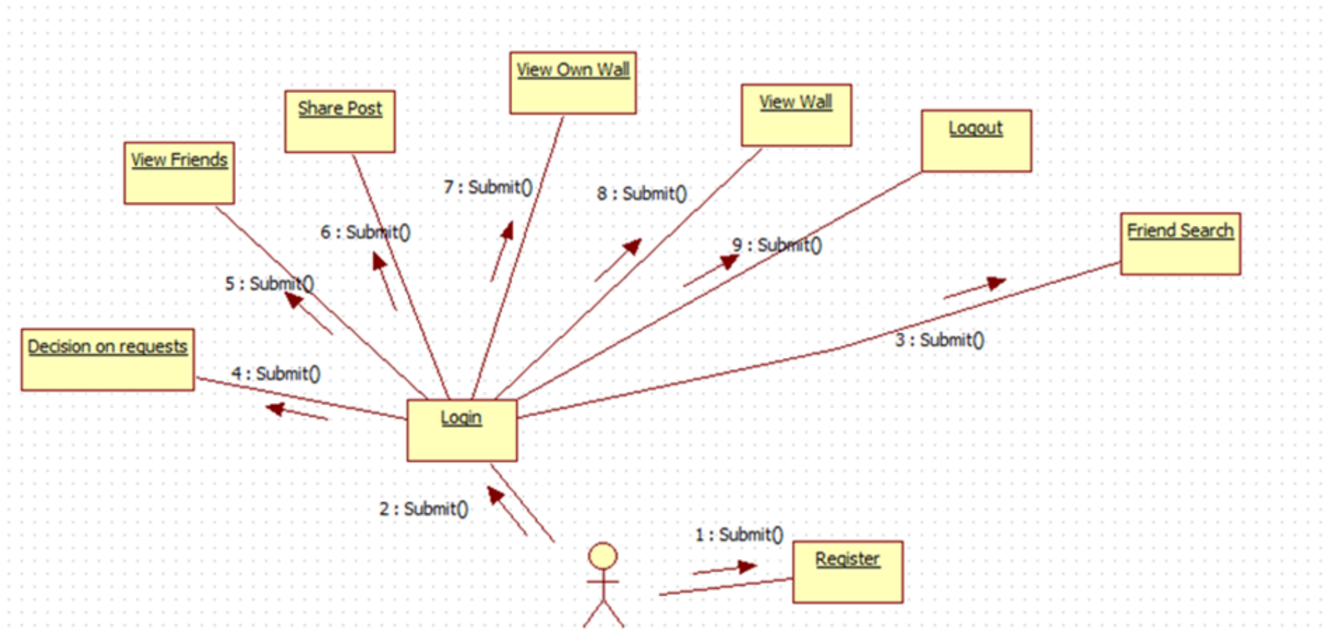
3.5 ADMIN COLLABORATION DIAGRAM



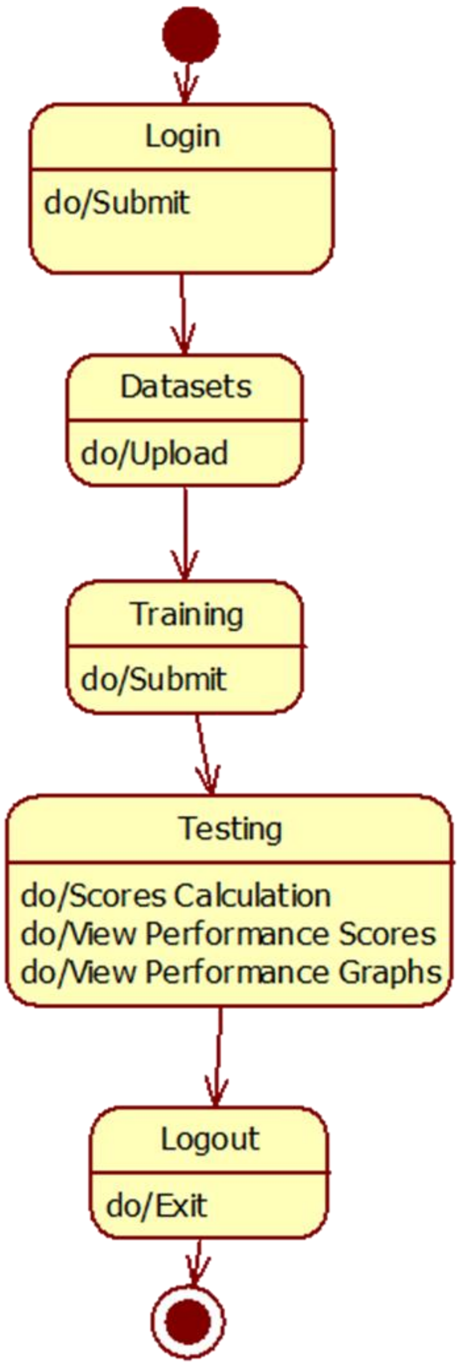
3.6 USER SEQUENCE



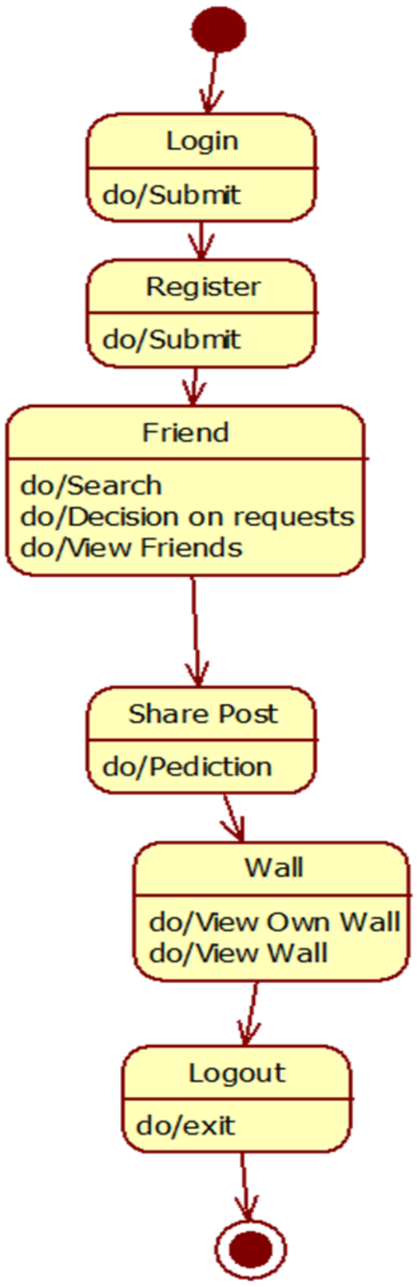
3.7 USER COLLABORATION DIAGRAM



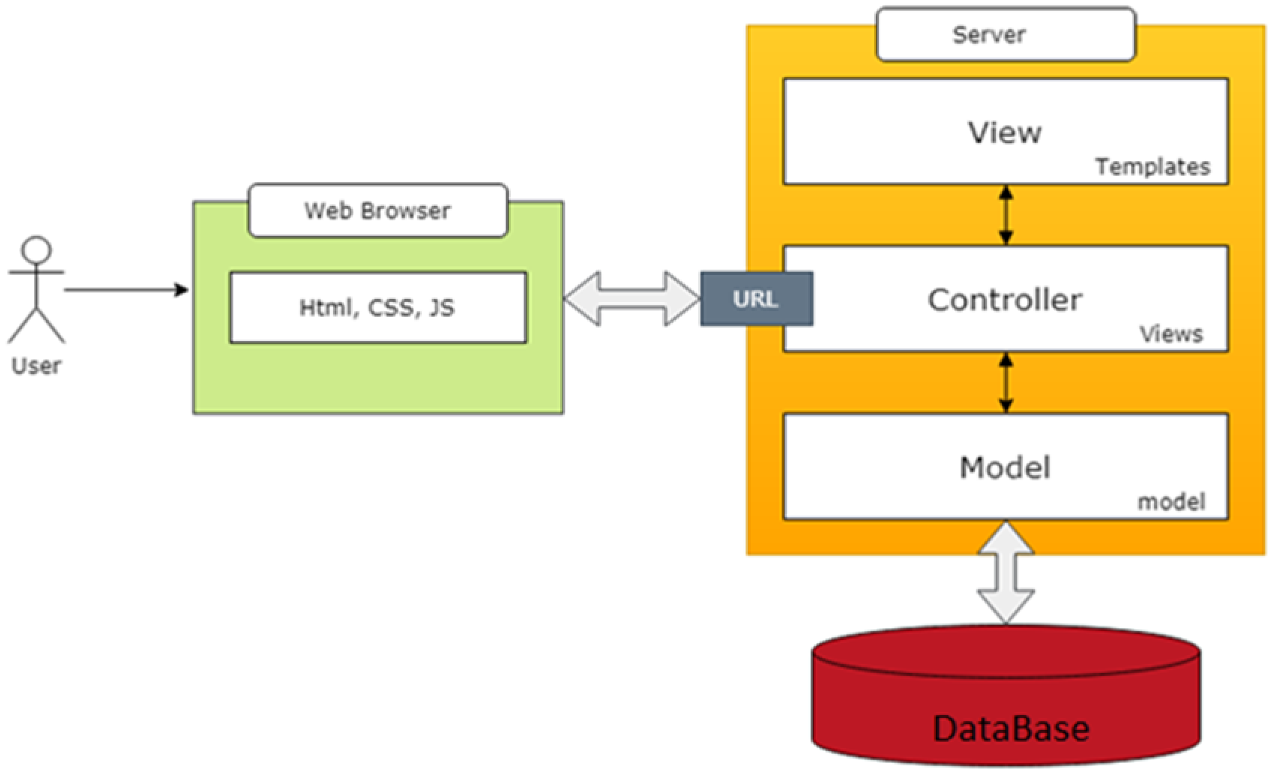
3.8 ADMIN START CHART



3.9 USER START CHART



3.10 DEPLOYMENT DIAGRAM



In the current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous user accounts are creating very rapidly. We need to focus on tracking the anonymous users. In our project we have calculated the user behavior according to the chat statements of the user which he/she does with others. By taking advantage of Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

4. IMPLEMENTATION

4.IMPLEMENTATION

ABOUT MYSQL:

MySQL is a relational database management system (RDBMS)[1] that runs as a server providing multi-user access to a number of databases.The SQL phrase stands for Structured Query Language.Free-software-open source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google , Facebook, and Twitter.

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's.

4.1 SAMPLE CODE:

```
from django.shortcuts import render

from django.http import HttpResponse, request

from .models import *

import matplotlib.pyplot as plt;

import numpy as np

import numpy

from django.shortcuts import render, redirect

from PIL import ImageTk, Image

from .Prediction import NN

from .Prediction2 import RF2

def homepage(request):

return render(request, 'index.html')

def signuppage(request):

if request.method=='POST':

e_mail=request.POST['mail']

d=usertab.objects.filter(e_mail__exact=e_mail).count()

if d>0:
```

```
return render(request, 'signup.html',{'msg':"e_mail Already Registered"})

else:

pass_word=request.POST['pass_word']

phone=request.POST['phone']

n_a_m_e=request.POST['n_a_m_e']

gen=request.POST['gen']

picture=request.POST['picture']

d=usertab(n_a_m_e=n_a_m_e,e_mail=e_mail,pass_word=pass_word,phone=phone,gender=gen,picture
=picture)

d.save()

return render(request, 'signup.html',{'msg':"Register Success, You can Login.."})

else:

return render(request, 'signup.html')

def userloginaction(request):

if request.method=='POST':

uid=request.POST['mail']

pass_word=request.POST['pass_word']

d=usertab.objects.filter(e_mail__exact=uid).filter(pass_word__exact=pass_word).count()

if d>0:

d=usertab.objects.filter(e_mail__exact=uid)
```

```
request.session['e_mail']=uid

request.session['n_a_m_e']=d[0].n_a_m_e

fc=frequest.objects.filter(to_e_mail__exact=uid).filter(stz__exact='request').count()

if fc>0:

fc=str(fc)+str(" new")

else:

fc=""

return render(request, 'user_home.html',{'data': d[0],'fc':fc})

else:

return render(request, 'user.html',{'msg':"Login Fail"})

else:

return render(request, 'user.html')

def adminloginaction(request):

if request.method == 'POST':

uid = request.POST['uid']

pwd = request.POST['pwd']

if uid == 'admin' and pwd == 'admin':

request.session['adminid'] = 'admin'

return render(request, 'admin_home.html')
```

```
else:

return render(request, 'admin.html', {'msg': "Login Fail"})

else:

return render(request, 'admin.html')

def adminhomedef(request):

if "adminid" in request.session:

uid = request.session["adminid"]

return render(request, 'admin_home.html')

else:

return render(request, 'admin.html')

def training(request):

if "adminid" in request.session:

uid = request.session["adminid"]

return render(request, 'training.html')

else:

return render(request, 'admin.html')

def testing(request):

if "adminid" in request.session:

uid = request.session["adminid"]
```

```
return render(request, 'testing.html')

else:

return render(request, 'admin.html')

def adminlogoutdef(request):

try:

del request.session['adminid']

except:

pass

return render(request, 'admin.html')

def userlogoutaction(request):

try:

del request.session['e_mail']

except:

pass

return render(request, 'user.html')

def userhomepage(request):

if "e_mail" in request.session:

e_mail=request.session["e_mail"]

d=usertab.objects.filter(e_mail__exact=e_mail)
```

```
fc=frequest.objects.filter(to_e_mail__exact=e_mail).filter(stz__exact='request').count()

print('..........',fc)

if fc>0:

fc=str(fc)+str(" new")

else:

fc=""

return render(request, 'user_home.html',{'data': d[0],'fc':fc})

else:

return redirect('n_userlogout')

def viewprofilepage(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]

d=usertab.objects.filter(e_mail__exact=uid)

return render(request, 'viewpprofile.html',{'data': d[0]})

else:

return render(request, 'user.html')

def fsearch(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]
```

```
fe_mail=request.POST['e_mail']

d=usertab.objects.filter(e_mail__exact=fe_mail)

if len(d)>0:

return render(request, 'fsearch.html',{'data': d[0]})

else:

return render(request, 'msg.html',{'msg': "No Details Available"})

else:

return render(request, 'user.html')

def freqsend(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]

un_a_m_e=request.session["n_a_m_e"]

fe_mail=request.POST['e_mail']

fn_a_m_e=request.POST['n_a_m_e']

d=frequest.objects.filter(fe_mail__exact=uid).filter(to_e_mail__exact=fe_mail).count()

if d>0:

d=usertab.objects.filter(e_mail__exact=uid)

return render(request, 'user_home.html',{'data': d[0],'msg':'Already sent friend request'})

else:
```

```
d=frequest(fn_a_m_e=un_a_m_e,fe_mail=uid,to_e_mail=fe_mail,stz='request')

d.save()

d=usertab.objects.filter(e_mail__exact=uid)

return render(request, 'user_home.html',{'data': d[0],'msg':'Friend Request Sent Successfully'})

else:

return render(request, 'user.html')

def viewfreq(request):

uid=request.session["e_mail"]

un_a_m_e=request.session["n_a_m_e"]

if request.method=='POST':

fe_mail=request.POST['e_mail']

fn_a_m_e=request.POST['n_a_m_e']

d=friends.objects.filter(e_mail__exact=uid).filter(frnd_e__exact=fe_mail).count()

if d>0:

return render(request, 'user_home.html',{'data': d[0],'msg':'Your Already Friends'})

else:

d=friends(e_mail=uid,frnd_e=fe_mail,frnd_n=fn_a_m_e)

d.save()

d=friends(e_mail=fe_mail,frnd_e=uid,frnd_n=un_a_m_e)
```

```
d.save()

frequest.objects.filter(to_e_mail = uid).filter(fe_mail = fe_mail).update(stz = 'accepted')

d=usertab.objects.filter(e_mail__exact=uid)

return render(request, 'user_home.html',{'data': d[0],'msg':'Updated !!'})

else:

d=frequest.objects.filter(to_e_mail__exact=uid).filter(stz__exact="request")

return render(request, 'viewfreq.html',{'data': d})

def reqreject(request):

uid=request.session["e_mail"]

fe_mail=request.POST['e_mail']

frequest.objects.filter(to_e_mail = uid).filter(fe_mail = fe_mail).update(stz = 'rejected')

d=usertab.objects.filter(e_mail__exact=uid)

return render(request, 'user_home.html',{'data': d[0],'msg':'Updated !!'})

def viewfrds(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]

d=friends.objects.filter(e_mail__exact=uid)

return render(request, 'viewfrds.html',{'data': d})

else:
```

```
return render(request, 'user.html')

def writepost(request):

if request.method=='POST':

import random

msg=request.POST['msg']

n_a_m_e=request.session["n_a_m_e"]

e_mail=request.session["e_mail"]

picture=request.POST['picture']

rs=NN.detecting(msg)

print('..........', rs)

if rs=='Non-offensive':

d=posts(n_a_m_e=n_a_m_e,e_mail=e_mail,msg=msg,picture=picture,stz='non',stz2='False')

d.save()

else:

rs=RF2.detecting(msg)

d=posts(n_a_m_e=n_a_m_e,e_mail=e_mail,msg=msg,picture=picture,stz=rs, stz2="True")

d.save()

return render(request, 'writepost.html',{'msg':"Post shared.."})

else:
```

```
return render(request, 'writepost.html')

def writepost2(request):

if request.method=='POST':

import random

msg=request.POST['msg']

n_a_m_e=request.session["n_a_m_e"]

e_mail=request.session["e_mail"]

picture='non'

rs=NN.detecting(msg)

print('..........', rs)

if rs=='Non-offensive':

d=posts(n_a_m_e=n_a_m_e,e_mail=e_mail,msg=msg,picture=picture,stz='non',stz2='False')

d.save()

else:

rs=RF2.detecting(msg)

d=posts(n_a_m_e=n_a_m_e,e_mail=e_mail,msg=msg,picture=picture,stz=rs, stz2="True")

d.save()

return render(request, 'writepost.html',{'msg':"Post shared.."})

else:
```

```
return render(request, 'writepost.html')

def ownwall(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]

d=posts.objects.filter(e_mail__exact=uid).order_by('-id')

return render(request, 'ownwall.html',{'data': d})

else:

return render(request, 'user.html')

def viewwall(request):

if "e_mail" in request.session:

uid=request.session["e_mail"]

d=posts.objects.all().order_by('-id')

r=[]

pp=True

for d1 in d:

pp=True

if d1.picture=='non':

pp=False

d3=friends.objects.filter(e_mail__exact=uid).filter(frnd_e__exact=d1.e_mail).count()
```

```
if d3>0:

ss=None

if d1.stz2=='True':

ss=True

ss2=False

else:

ss=False

ss2=True

print('>>>>>>>>>>>',bool(d1.stz2))

r.append({'n':d1.n_a_m_e,'p':d1.picture,'m':d1.msg,'stz1':d1.stz,'stz2':ss,'stz3':ss2,'pstz':pp})

return render(request, 'viewwall.html',{'data': r})

else:

return render(request, 'user.html')

def d1svmdef(request):

from .D1SVM import model

model()

return render(request, 'training.html', {'msg': "SVM Classifier Training Completed Successfully"})

def d1nbdef(request):

from .D1NB import model
```

```
model()

return render(request, 'training.html', {'msg': "Naive Bayees Classifier Training Completed Successfully"})

def d1nndef(request):

from .D1NN import model

model()

return render(request, 'training.html', {'msg': "Neural Network Classifier Training Completed Successfully"})

def d1rfdef(request):

from .D1RF import model

model()

return render(request, 'training.html', {'msg': "Random Forest Classifier Training Completed Successfully"})

def d2rfdef(request):

from .D2RF import model

model()

return render(request, 'training.html', {'msg': "Random Forest Classifier Training Completed Successfully"})

def d2svmdef(request):

from .D2SVM import model

model()
```

```
return render(request, 'training.html', {'msg': "SVM Classifier Training Completed Successfully"})

def d2nbdef(request):

from .D2NB import model

model()

return render(request, 'training.html', {'msg': "Naive Bayees Classifier Training Completed Successfully"})

def d2nndef(request):

from .D2NN import model

model()

return render(request, 'training.html', {'msg': "Neural Network Classifier Training Completed Successfully"})

def d1testingdef(request):

from .D1Testing import D1Testing

D1Testing.main()

return render(request, 'testing.html', {'msg': "Testing of dataset1 completed.."})

def d2testingdef(request):

from .D2Testing import D2Testing

D2Testing.main()

return render(request, 'testing.html', {'msg': "Testing of dataset2 completed.."})

def results(request):
```

```
if "adminid" in request.session:

d = performance.objects.all()

return render(request, 'viewaccuracy.html', {'data': d})

else:

return render(request, 'admin.html')

def viewgraph(request, cat):

if "adminid" in request.session:

algorithms = ['NB D1','SV M D1','NN D1','RF D1', 'NB D2','SV M D2','NN D2','RF D2']

plt.cla()

plt.clf()

row = performance.objects.all()

rlist = []

for r in row:

if cat == 'acc_v':

rlist.append(float(r.acc))

plt.title('Accuracy Measure')

elif cat == 'pre_v':

rlist.append(float(r.prec))

plt.title('Precision Measure')
```

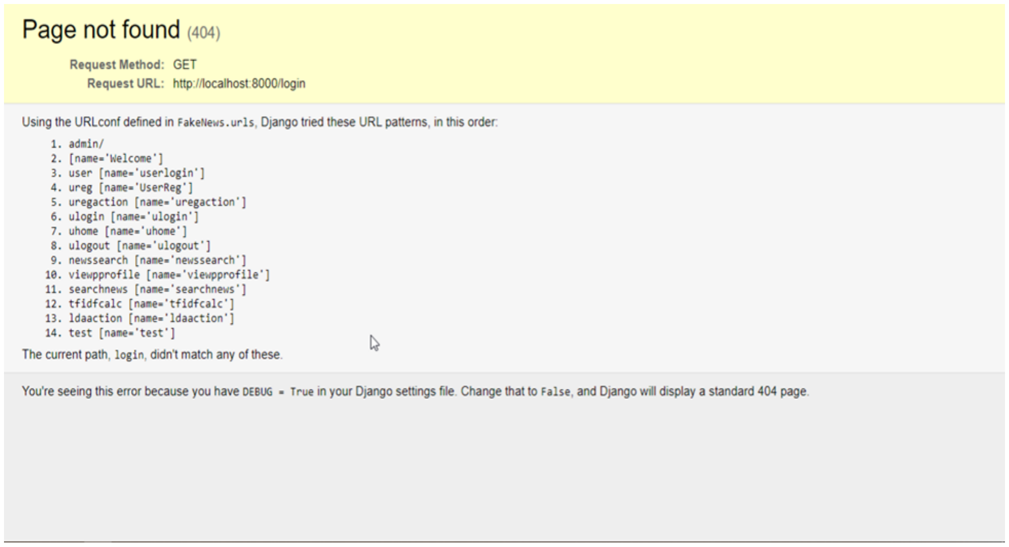
```
elif cat == 'rec_v':

rlist.append(float(r.recall))

plt.title('Recall Measure')

elif cat == 'f1_v':

rlist.append(float(r.f1))

plt.title('F1-Score Measure')

try:

height = rlist

plt.xlabel('')

plt.ylabel('Algorithms ')

from PIL import Image

plt.savefig(str(cat)+'.jpg')

except Exception as e:

print(e)

from PIL import Image

print(''+str(cat)+''+'.jpg','<<<<<<<<<<<<<<<<<<<')

im = Image.open(str(cat)+'.jpg')

im.show()

return redirect('results')
```

5.TESTING

5.TESTING:

5.1 URL Mismatch Error:

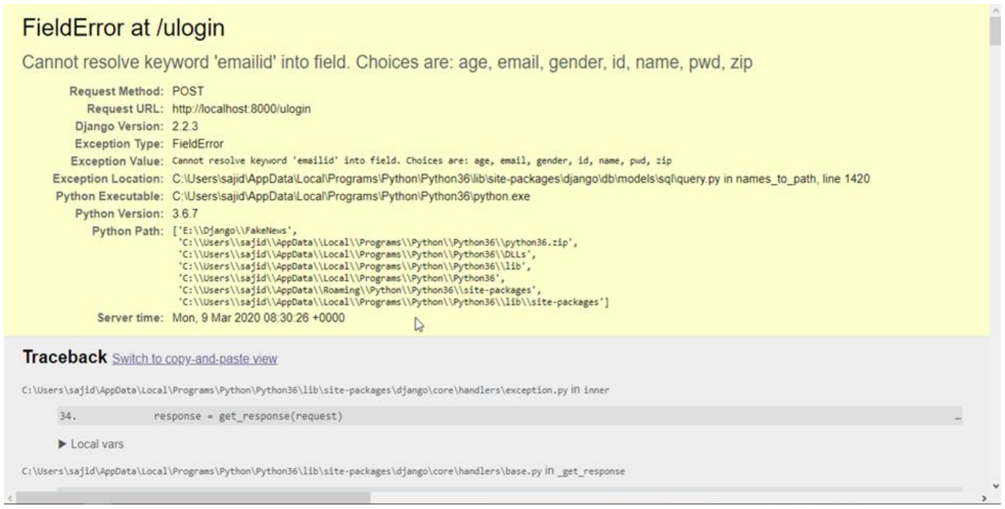
When we give a URL like localhost:8000/login and if it’s not match in urls.py files, we can get this error.



```
Page not found (404)
        Request Method:  GET
           Request URL:  http://localhost:8000/login

Using the URLconf defined in FakeNews.urls, Django tried these URL patterns, in this order:
    1. admin/
    2. [name='Welcome']
    3. user [name='userlogin']
    4. ureg [name='UserReg']
    5. uregaction [name='uregaction']
    6. ulogin [name='ulogin']
    7. uhome [name='uhome']
    8. ulogout [name='ulogout']
    9. newssearch [name='newssearch']
   10. viewpprofile [name='viewpprofile']
   11. searchnews [name='searchnews']
   12. tfidfcalc [name='tfidfcalc']
   13. ldaaction [name='ldaaction']
   14. test [name='test']
The current path, login, didn't match any of these.

You're seeing this error because you have DEBUG = True in your Django settings file. Change that to False, and Django will display a standard 404 page.
```

5.2 Field Error

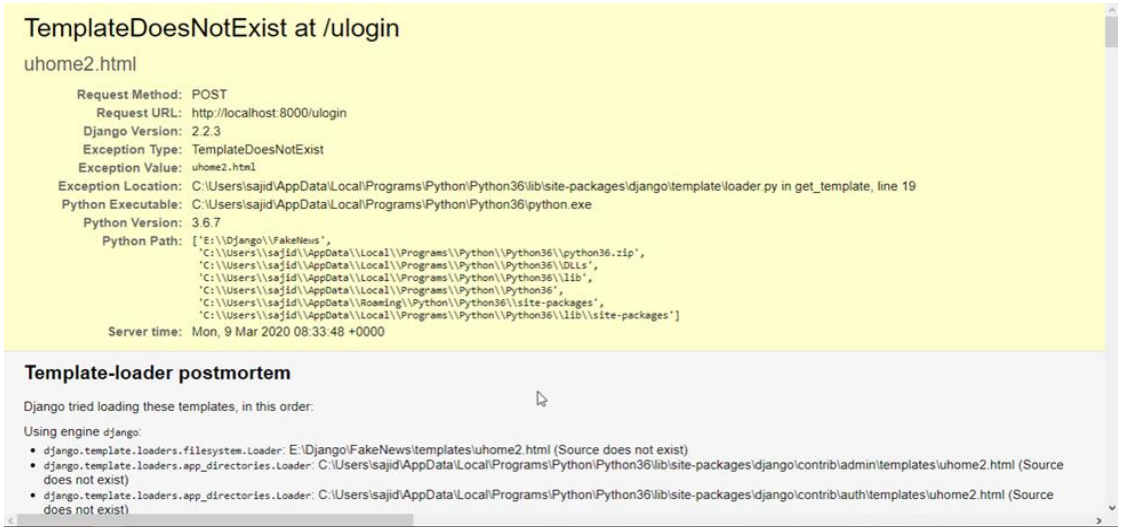
Database field mismatch from model. Given keyword 'emailid' into field. expected: age, email, gender, id, name, pwd, zip.



```
FieldError at /ulogin
Cannot resolve keyword 'emailid' into field. Choices are: age, email, gender, id, name, pwd, zip
      Request Method:  POST
         Request URL:  http://localhost:8000/ulogin
      Django Version:  2.2.3
      Exception Type:  FieldError
     Exception Value:  Cannot resolve keyword 'emailid' into field. Choices are: age, email, gender, id, name, pwd, zip
  Exception Location:  C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\db\models\sql\query.py in names_to_path, line 1420
   Python Executable:  C:\Users\sajid\AppData\Local\Programs\Python\Python36\python.exe
      Python Version:  3.6.7
         Python Path:  ['E:\\Django\\FakeNews',
                        'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\python36.zip',
                        'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\DLLs',
                        'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\lib',
                        'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36',
                        'C:\\Users\\sajid\\AppData\\Roaming\\Python\\Python36\\site-packages',
                        'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\lib\\site-packages']
         Server time:  Mon, 9 Mar 2020 08:30:26 +0000

Traceback Switch to copy-and-paste view
C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\handlers\exception.py in inner
    34.            response = get_response(request)
    ▶ Local vars
C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\handlers\base.py in _get_response
```

5.3 Template Does Not Exist

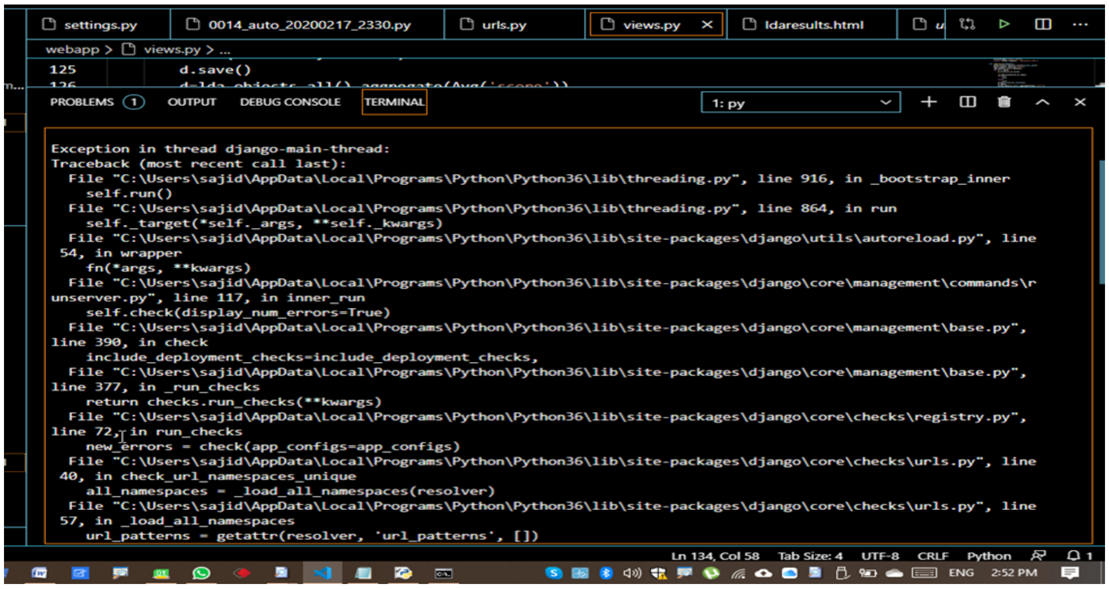
We get this error when URL redirecting and template file (html) not found, or mismatch



```
TemplateDoesNotExist at /ulogin
uhome2.html

       Request Method:  POST
          Request URL:  http://localhost:8000/ulogin
       Django Version:  2.2.3
       Exception Type:  TemplateDoesNotExist
      Exception Value:  uhome2.html
   Exception Location:  C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\template\loader.py in get_template, line 19
    Python Executable:  C:\Users\sajid\AppData\Local\Programs\Python\Python36\python.exe
       Python Version:  3.6.7
          Python Path:  ['E:\\Django\\FakeNews',
                         'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\python36.zip',
                         'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\DLLs',
                         'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\lib',
                         'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36',
                         'C:\\Users\\sajid\\AppData\\Roaming\\Python\\Python36\\site-packages',
                         'C:\\Users\\sajid\\AppData\\Local\\Programs\\Python\\Python36\\lib\\site-packages']
          Server time:  Mon, 9 Mar 2020 08:33:48 +0000

Template-loader postmortem

Django tried loading these templates, in this order:

Using engine django:
 • django.template.loaders.filesystem.Loader: E:\Django\FakeNews\templates\uhome2.html (Source does not exist)
 • django.template.loaders.app_directories.Loader: C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\contrib\admin\templates\uhome2.html (Source
   does not exist)
 • django.template.loaders.app_directories.Loader: C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\contrib\auth\templates\uhome2.html (Source
   does not exist)
```

5.4 Exception while running server

If any syntactically mistakes in views.py we get this exception.



```
        d.save()
```

```
Exception in thread django-main-thread:
Traceback (most recent call last):
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\threading.py", line 916, in _bootstrap_inner
    self.run()
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\threading.py", line 864, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\utils\autoreload.py", line
 54, in wrapper
    fn(*args, **kwargs)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\management\commands\r
unserver.py", line 117, in inner_run
    self.check(display_num_errors=True)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\management\base.py",
line 390, in check
    include_deployment_checks=include_deployment_checks,
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\management\base.py",
line 377, in _run_checks
    return checks.run_checks(**kwargs)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\checks\registry.py",
line 72, in run_checks
    new_errors = check(app_configs=app_configs)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\checks\urls.py", line
 40, in check_url_namespaces_unique
    all_namespaces = _load_all_namespaces(resolver)
  File "C:\Users\sajid\AppData\Local\Programs\Python\Python36\lib\site-packages\django\core\checks\urls.py", line
 57, in _load_all_namespaces
    url_patterns = getattr(resolver, 'url_patterns', [])
```

6.RESULTS

6.RESULTS:

6.1 User's Account Creation:

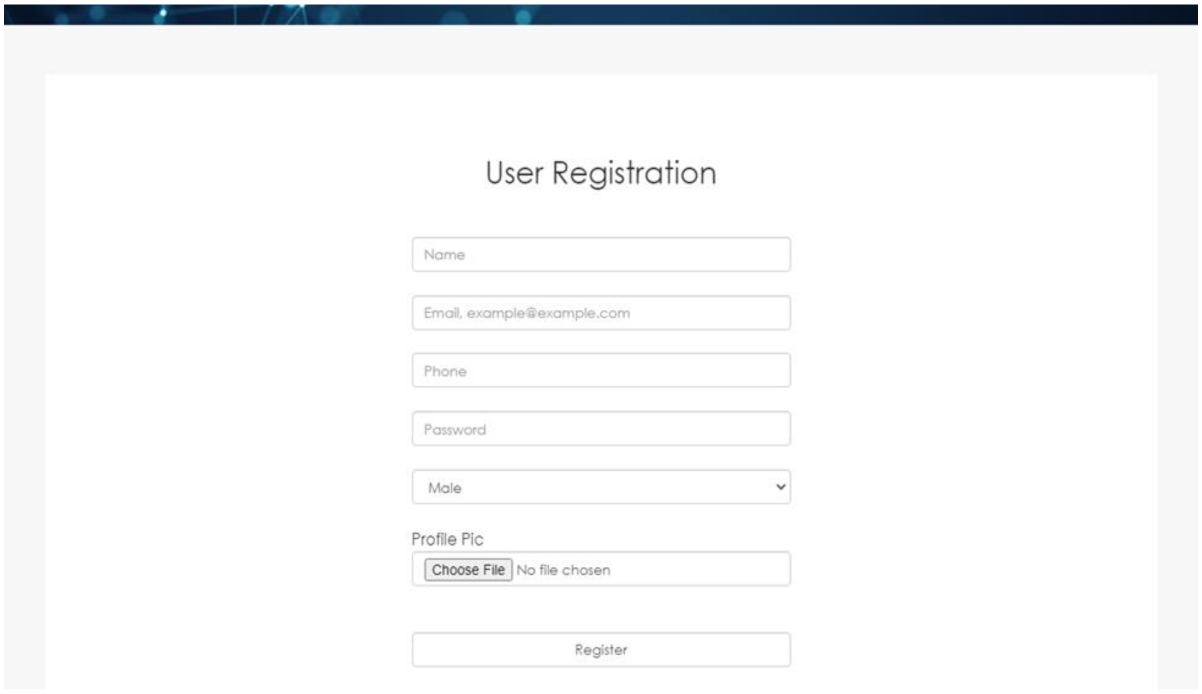


Figure 6.1 User Signup Page

6.2 Login page for user verification:

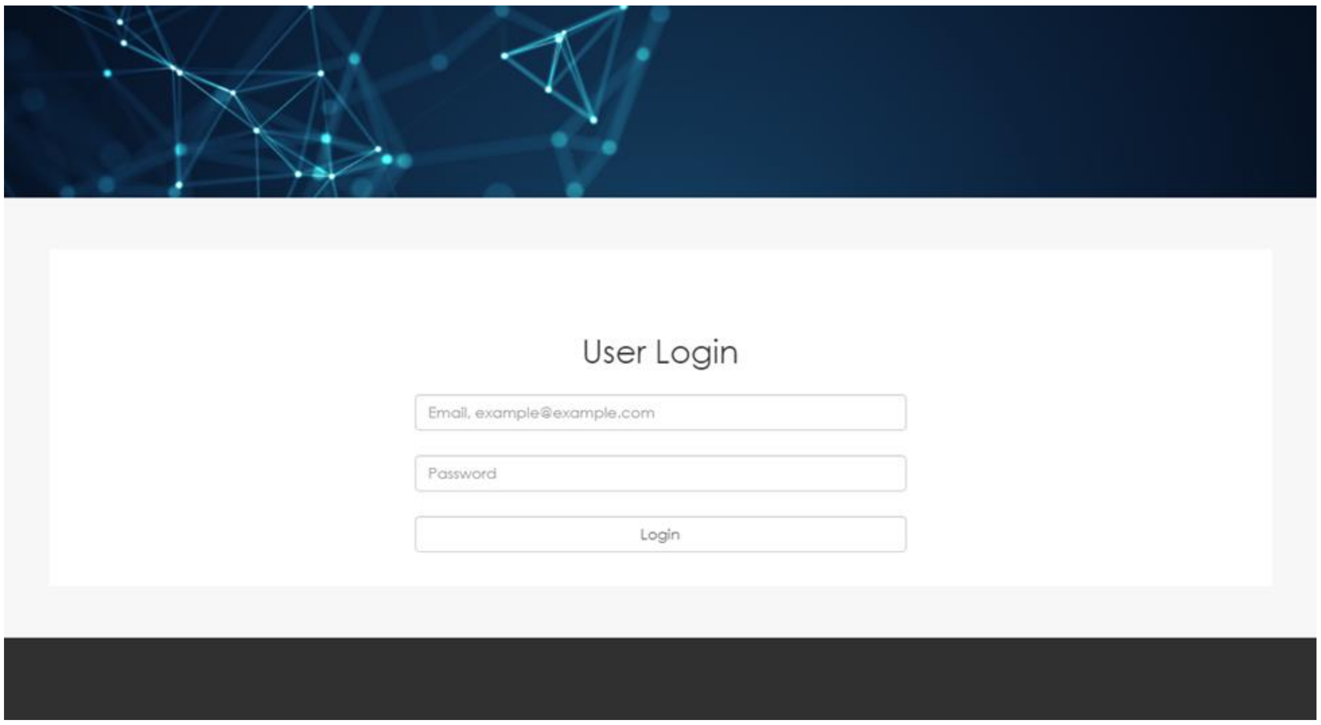


Figure 6.2 User Login Page

6.3 User Homepage

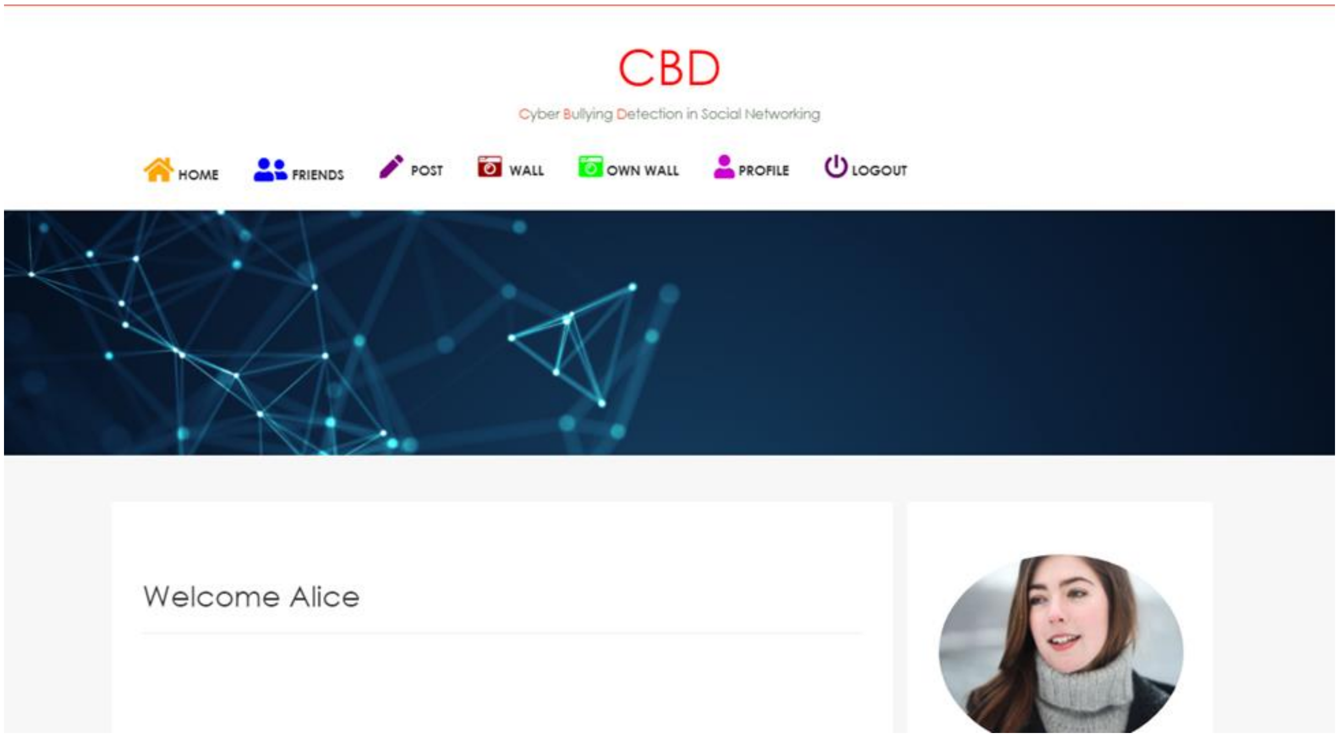


Figure 6.3 User Homepage

6.4 Search and Send Friend Requests

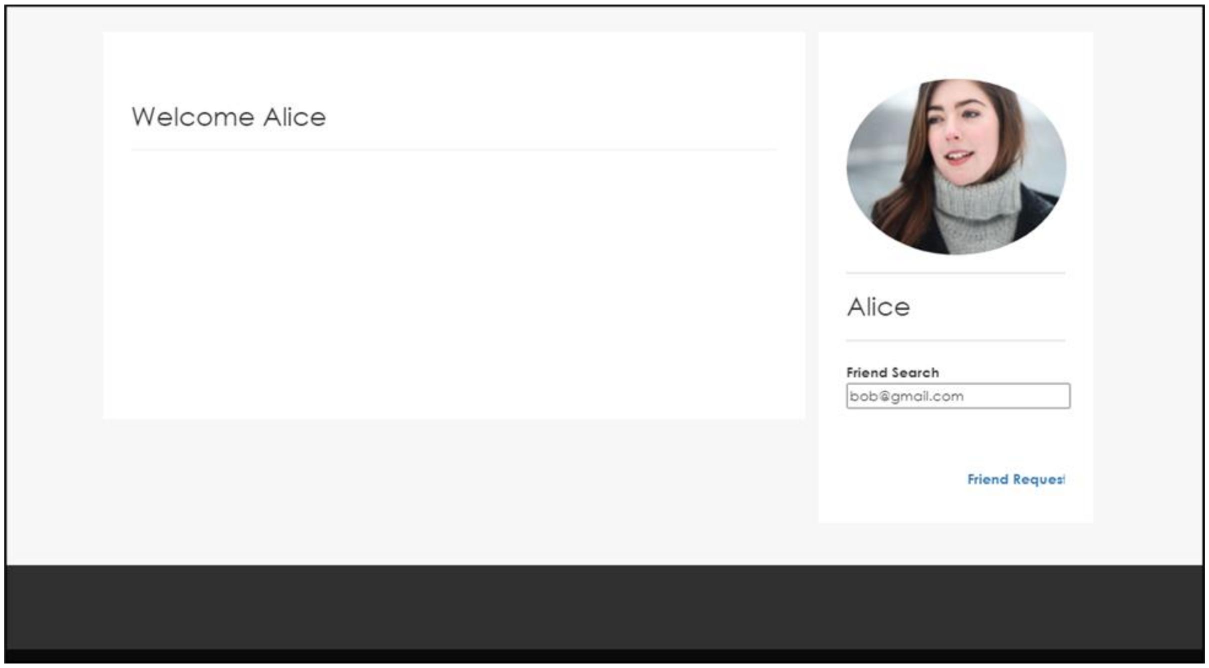
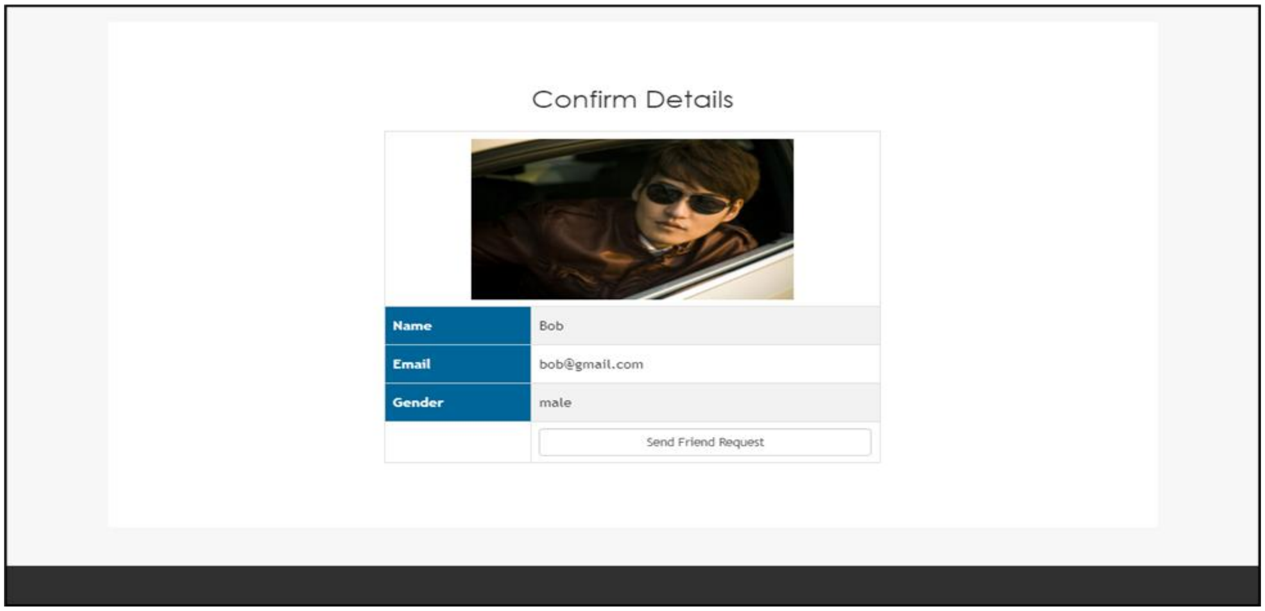


Figure 6.4: Search Friends option



Name	Bob
Email	bob@gmail.com
Gender	male

Figure 6.5: Send Friend request

6.5 View Friend Requests and Decision on requests

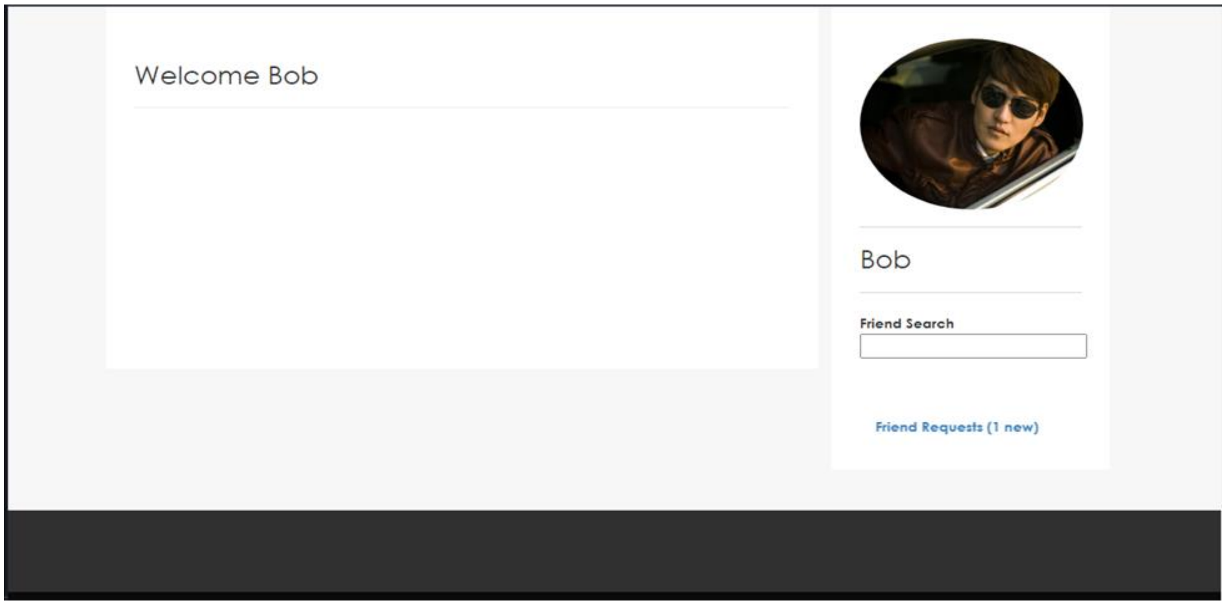


Figure 6.6: View Friend request

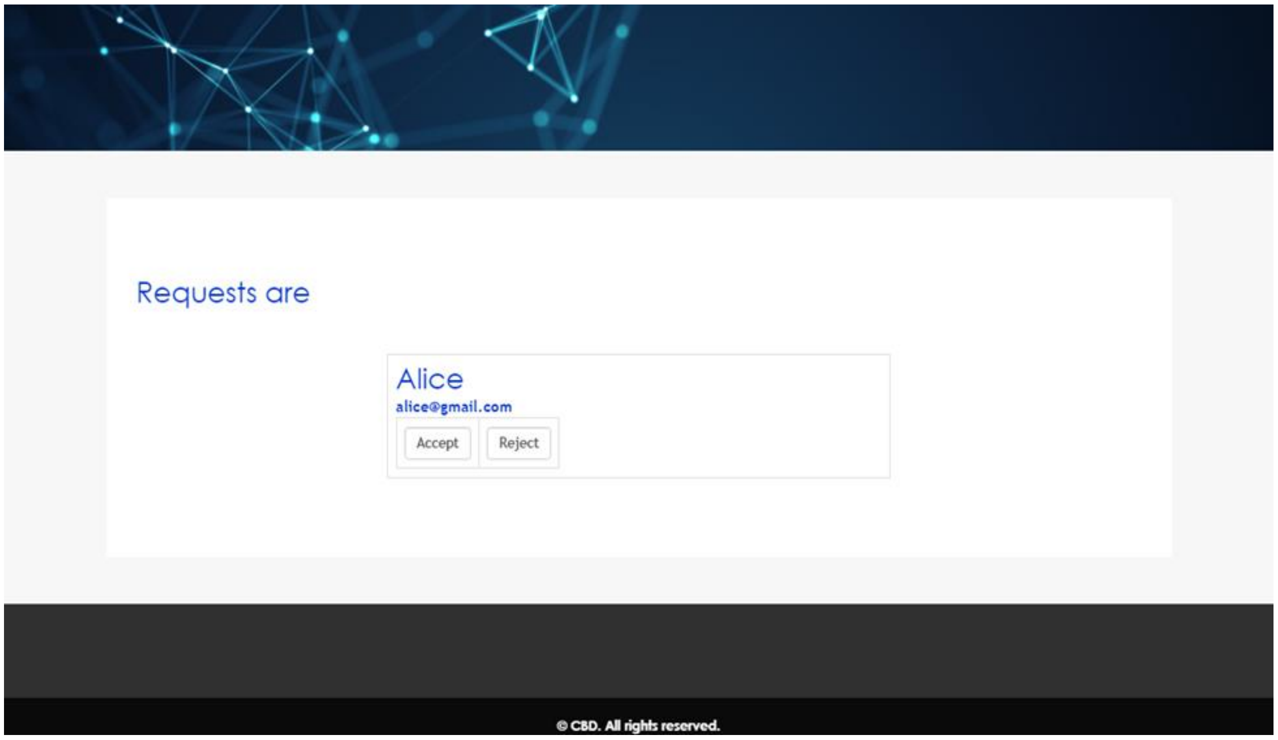
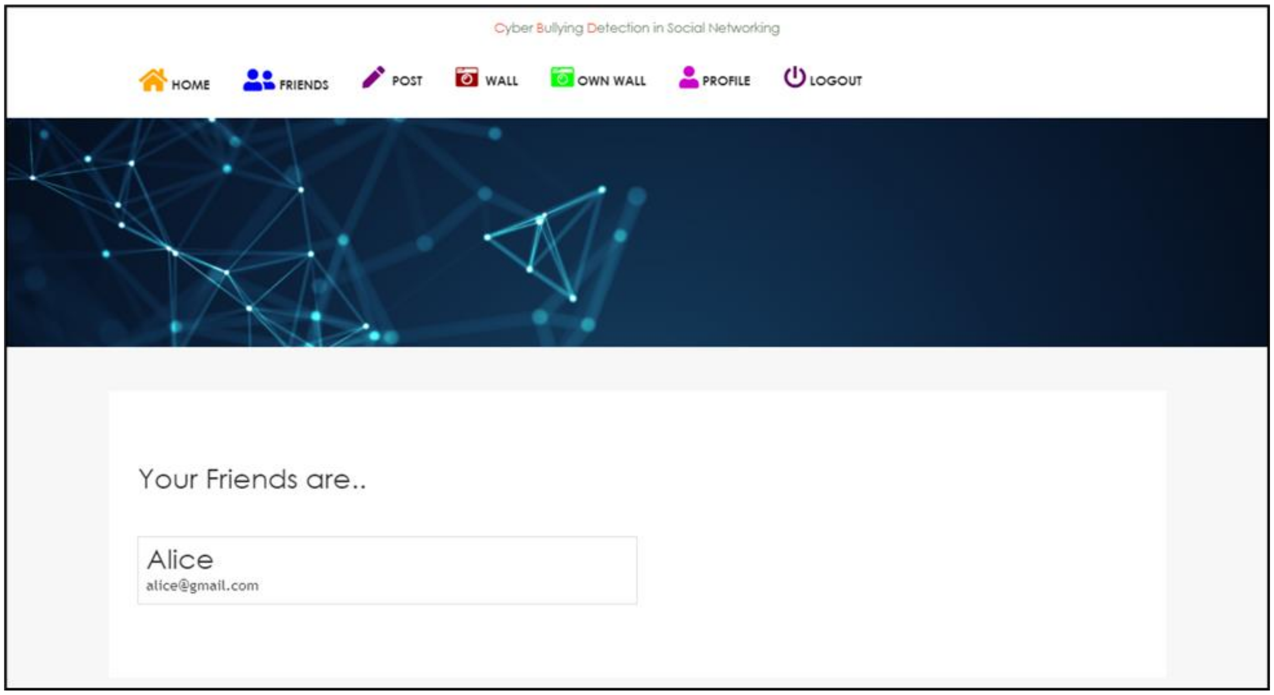


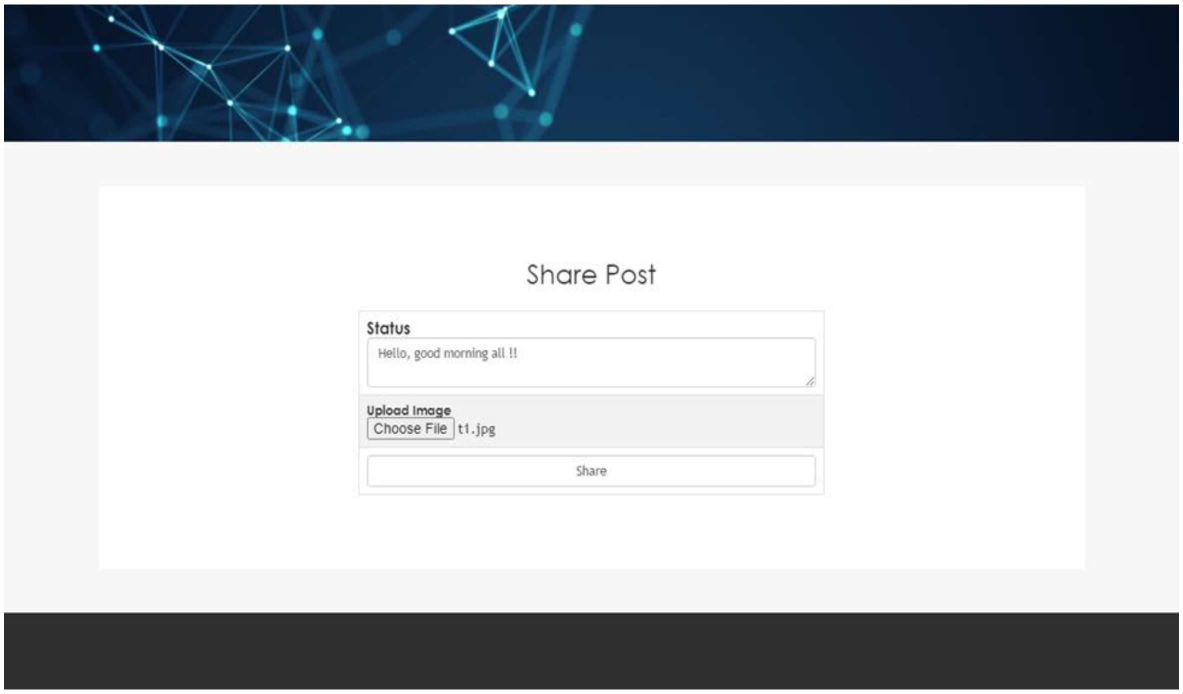
Figure 6.7: Decision on friend request



Your Friends are..

Alice
alice@gmail.com

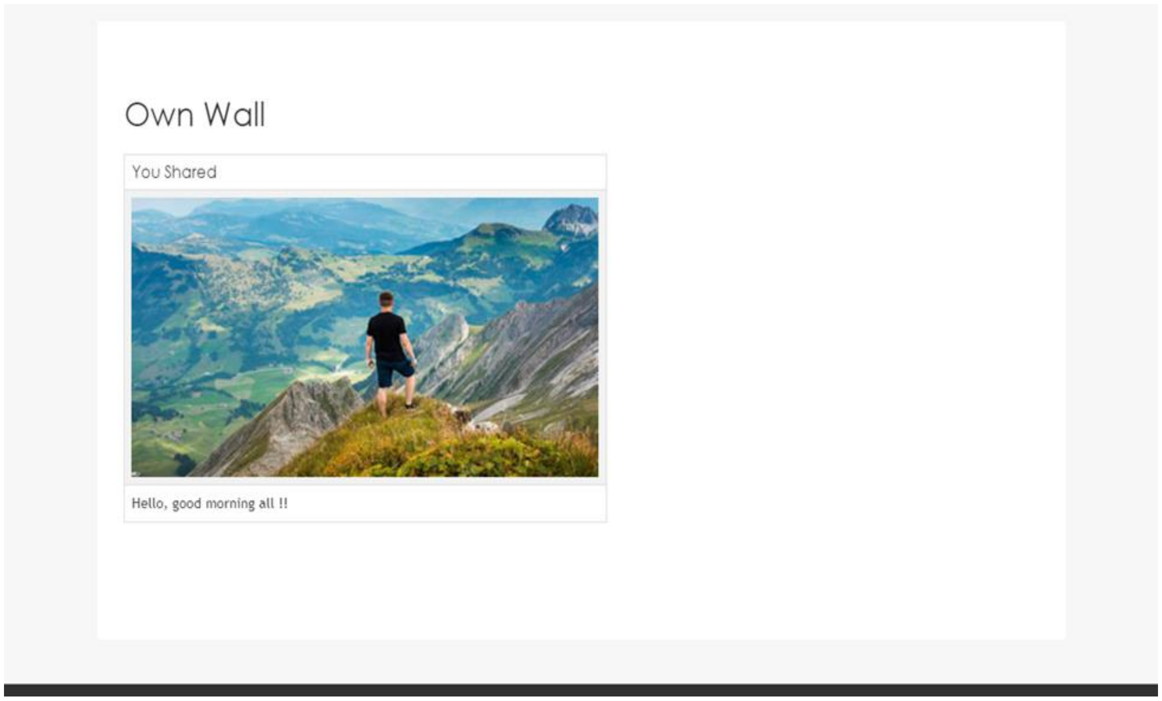
Figure 6.8: Decision on friends



Hello, good morning all !!

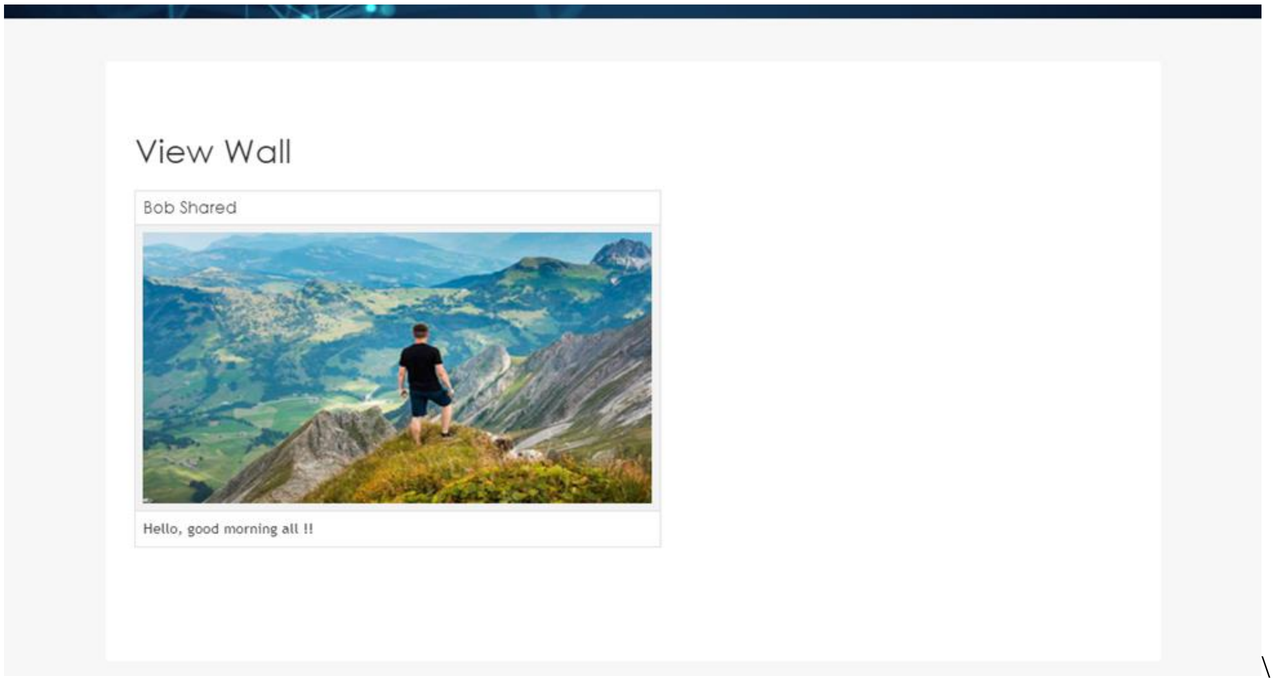
t1.jpg

Figure 6.9: Post Share



Hello, good morning all !!

Figure 6.10: View Own Wall



Hello, good morning all !!

\

Figure 6.11: View Wall

7.CONCLUSION

7.CONCLUSION:

In current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous user accounts are being created very rapidly. We need to focus on tracking the anonymous users. In our project we have calculated the user behavior according to the chat statements of the user which he/she does with others. By taking advantage of Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

8.BIBLIOGRAPHY

8.BIBLIOGRAPHY

8.1 REFERENCES:

[1] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, “Identifying users across social tagging systems,” in Proc. 5th Int. AAAI Conf. Weblogs Social Media, 2011, pp. 522–525.

[2] Identifying Users Across Social Tagging Systems Tereza Iofciu, Péter Fankhauser, +1 author Kerstin Bischoff, prieto et al. 2009

[3] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manila, “How unique and traceable are usernames?” in Proc. 11th Int. Conf. Privacy Enhancing Technology., 2011, pp. 1–17.

[4] J. Liu, F. Zhang, X. Song, Y. I. Song, C. Y. Lin, and H. W. Hon, “What’s in a name?: An unsupervised approach to link users across communities,” in Proc. 6th ACM Int. Conf. Web Search Data Mining, 2013, pp. 495–504.

8.2 GITHUB LINK

https://github.com/shanmukanjali/Detection-of-cyberbullying-on-social-media-using-machine-learning

9.PUBLICATION

© April 2023| IJIRT | Volume 9 Issue 11 | ISSN: 2349-6002

Detection of Cyberbullying on social media Using Machine Learning

Munnelli Shanmukanjali[1], Dharpalli Arun Praneeth Kumar[2], Pullareddy Varsheek Reddy[3], Dr.T.S.Mastan Rao[4]

[1,2,3]*B.Tech Student, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad, Telangana, India*

[4]*Associate Professor, Department of Computer Science and Engineering, CMR Technical Campus, Medchal, Hyderabad Telangana, India*

Abstract- In the realm of online social networks, it is crucial to conduct research on the detection of anonymous user behavior and offensive content. This particular project focuses on detecting bully statements and offensive data in shared content of social networks. To achieve accurate results, the project proposes a system called "Cyber Bullying Detection (CBD) in Social Networking," which utilizes Machine Learning algorithms and Text Mining concepts. The project employs two datasets, namely the 'Hate Speech and Offensive Language Dataset' and 'Harassment-Corpus Dataset,' and utilizes three Machine Learning classifiers, including Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), and Neural Network (NN) Algorithms to compare their performance on both datasets. The project also includes the design and development of a Python-based Django web application to demonstrate the system's results.

1.INTRODUCTION

Social networking apps, including chat applications such as WhatsApp and Messenger, provide a range of communication tools for text, media, and web data sharing. In recent years, social networking sites have expanded to offer extensive services such as multimedia and e-commerce. For instance, Twitter is a major platform for micro-blogging, with over 700 million users and 400 million micro-blogs produced daily. However, research has found that over 30% of accounts on social media services like Twitter, Facebook, and Sina are fake or duplicates. Despite this, social sites currently do not prioritize tracking the anonymous behavior of users.

Detecting malicious users has become a significant topic in the study of social media, as current social network sites do not prioritize tracking user behavior, especially that of anonymous users, and the practicality of implementing concepts like profile matching or network-based techniques in real-time is limited. Additionally, crawling user information from microblogs is impractical, and anonymous users can easily manipulate public profile information. Chat applications and multimedia data sharing (e.g., images, videos) are prevalent means of communication among social networking sites, with Facebook and Instagram serving as prime examples, but they also pose a security threat such as cyberbullying.

2. RELATED WORK

In the current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. For example twitter social media provides major services of micro-blogging massively, it has more than 700 million users and 400 million micro-blogs produced per day. According to a research survey, more than 30% of dummy or duplicate or fake accounts are present in all social media services like twitter, facebook, sina etc [1]. But the current social sites do not focus on services like tracking the user behavior or anonymous behavior. In the current system, social network sites need to focus on the user microblogs and need to capture the user behavior whether he/she is an anonymous user or not.

Few surveys' providing concepts to tracking the attackers like using profile matching techniques and network based techniques etc. But in real-time, applying those concepts in social networks is less practical. Crawling the user information from the user micro blogs is also less practical. Anonymous Users can easily manipulate the public profile information.

In the current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous user accounts are being created very rapidly. We need to focus on tracking the anonymous users. In our proposed system we are implementing a web based application which will find the anonymous users according to the user behavior. We calculate the user behavior according to the chat statements of the user which he/she does with others. By taking advantage of Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

3.PROPOSED SYSTEM

The proposed system 'Cyberbullying Detection (CBD) in Social Networking' is to identify the bully or offensive statements using a classification model. Based on this requirement, the proposed architecture is designed based on two flows, namely, classification analysis and user side prediction. Based on the project requirement, this architecture is designed which is represented in figure 1. This section described the workflow of the architecture and project main modules. The main purpose of the CBD system is to identify the bully or offensive statements using Machine Learning algorithms. To achieve this requirement, we need to implement classification analysis. The classification analysis is the process of conducting training and testing processes for calculating the performance measures between various Machine Learning algorithms. Based on these requirements, the proposed architecture is designed with two flows, namely, classification analysis which is taken care of by the admin and user side prediction. In the architecture these two flows are represented in two different color formats.

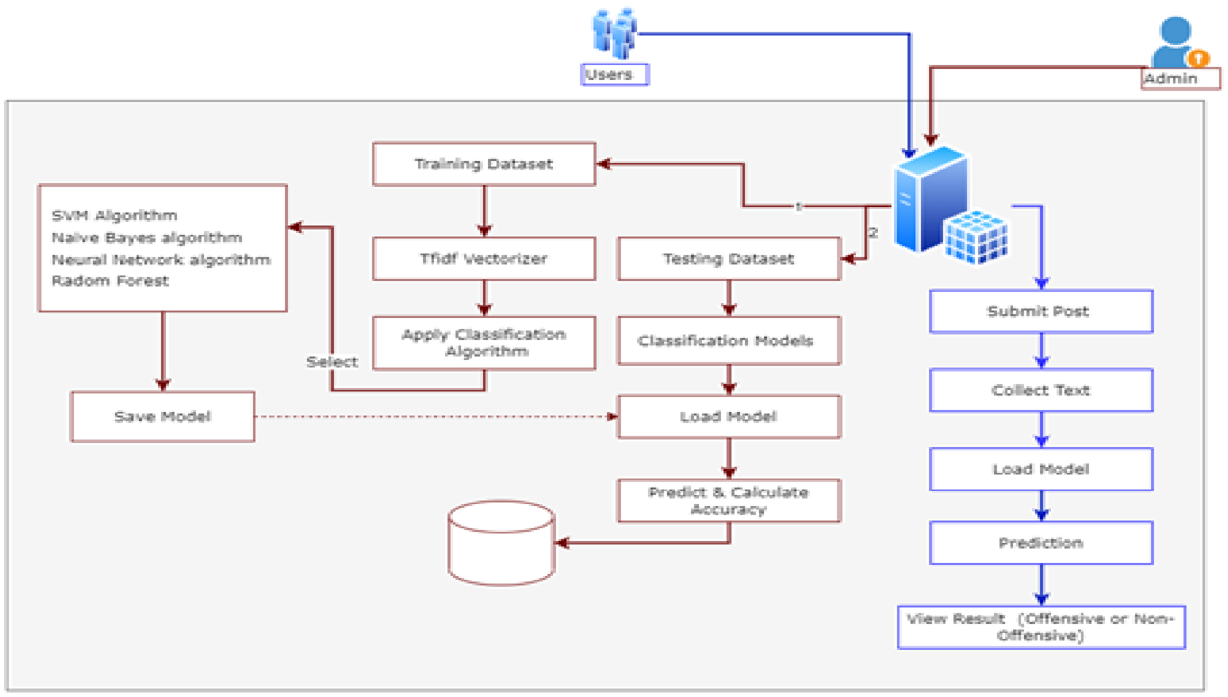


Fig:-1 CBD Architecture

Advantages:
We are depending on the chat history instead of the profile attributes or any other media attributes, so we can conclude with a minimum amount of the computation.
1. Due to machine learning classifications we can get accurate results
2. Cyberbullying detection process is automatic and time taken for detection is less and it works on the live environment.
3. The latest machine learning models are used for training models that are accurate.

Disadvantages:

1.Very less practical, we can't find the attacker using small tiny blogs
2.Based on the network based models we should effort heavy data for detecting
Modules:

Admin:
Admin is a main user of our application, admin will process main functionalities of the system. Admin can see the lists of users are available in the system. Admin can process the detection of anomaly users with Naïve Bayes algorithm. Admin will send a warning mail to the user. After that user if continue the same manner then admin will block the account of the anomaly user.

User:
In our system user is an end user of our application. We build a social application for users. They can make friends and share data among them. User also chat with his/her friends. When a user sends any bullying words to the others application will detect the user and send the warning mail to the user. User will get the notification of warning through mail. After getting the warning alert if user will continue the same manner then user will blocked in the system.

4.EXPERIMENTAL RESULTS

User's Account Creation:

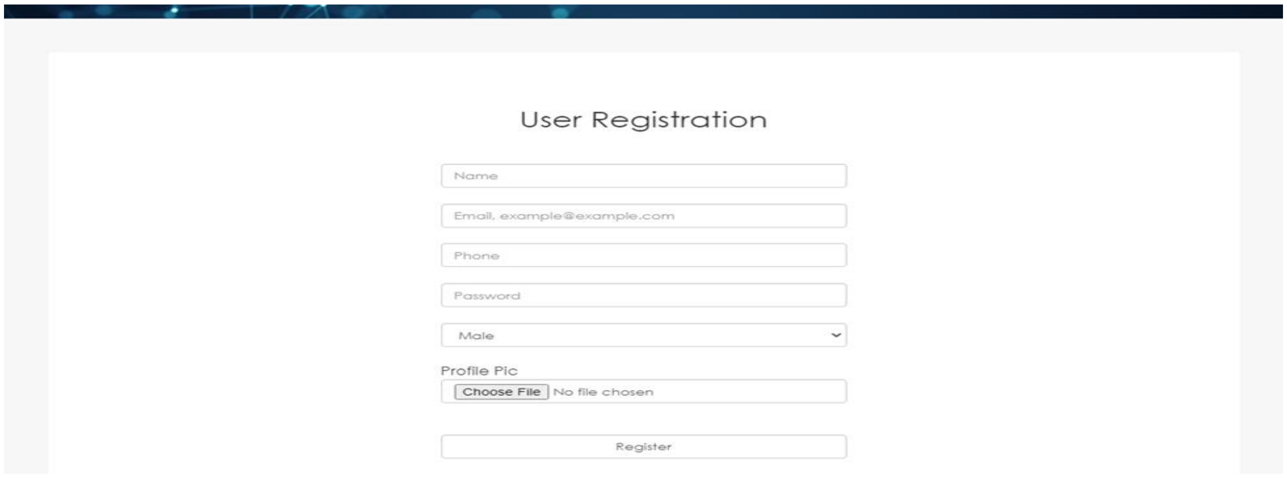


Figure 6.1 User Signup Page

Login page for user verification:

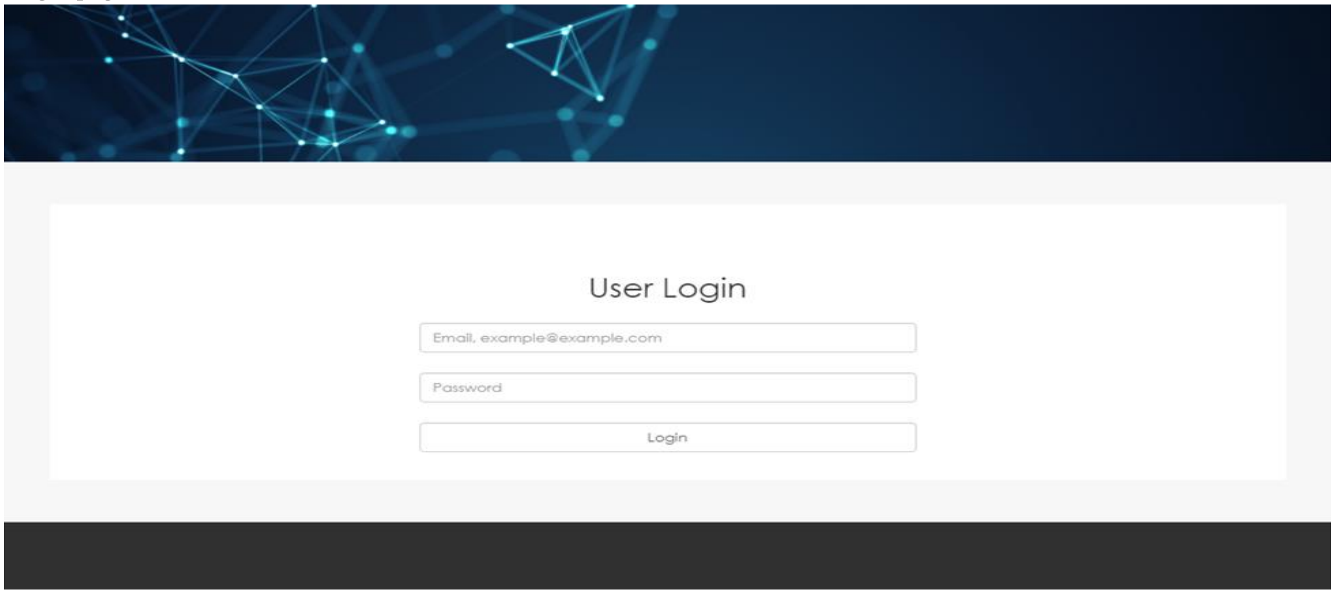


Figure 6.2 User Login Page

User Homepage

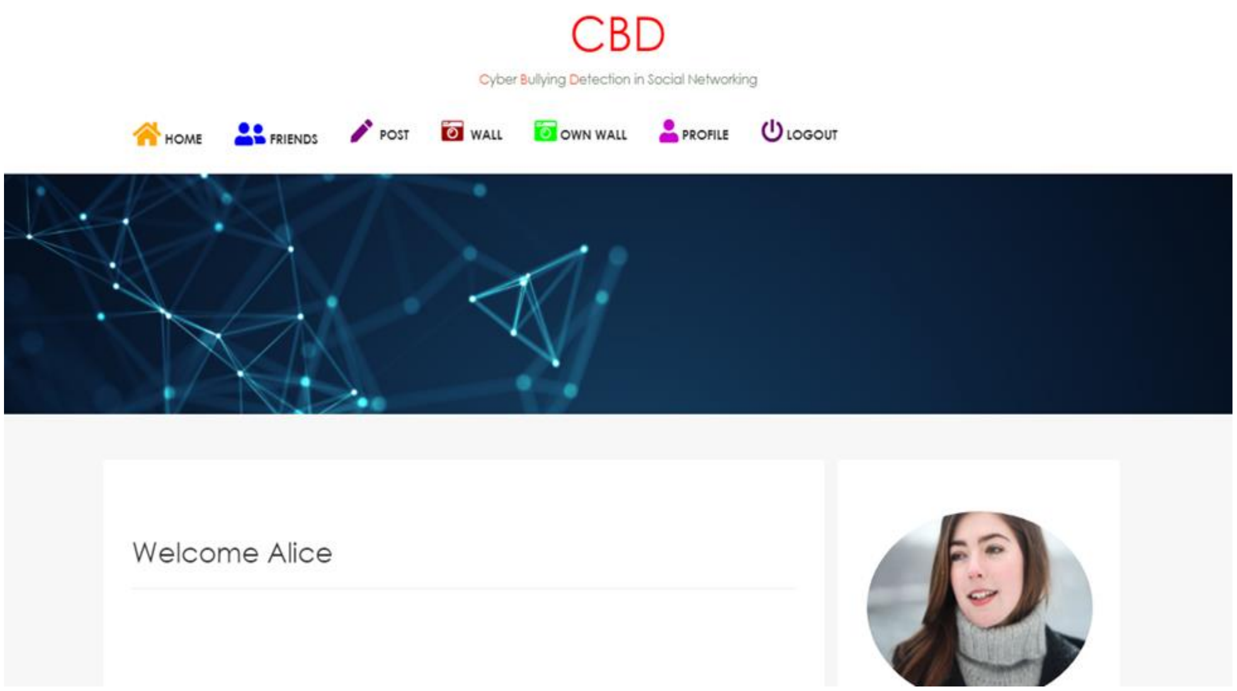


Figure 6.3 User Homepage

Search and Send Friend Requests

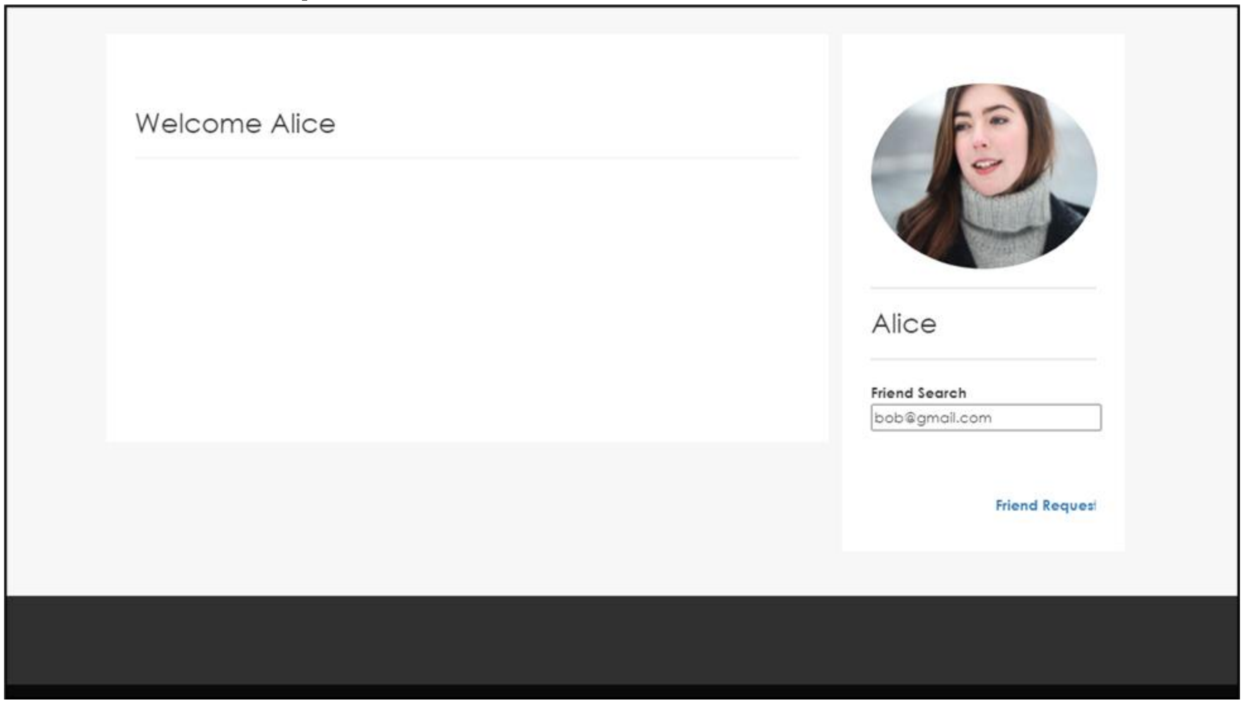
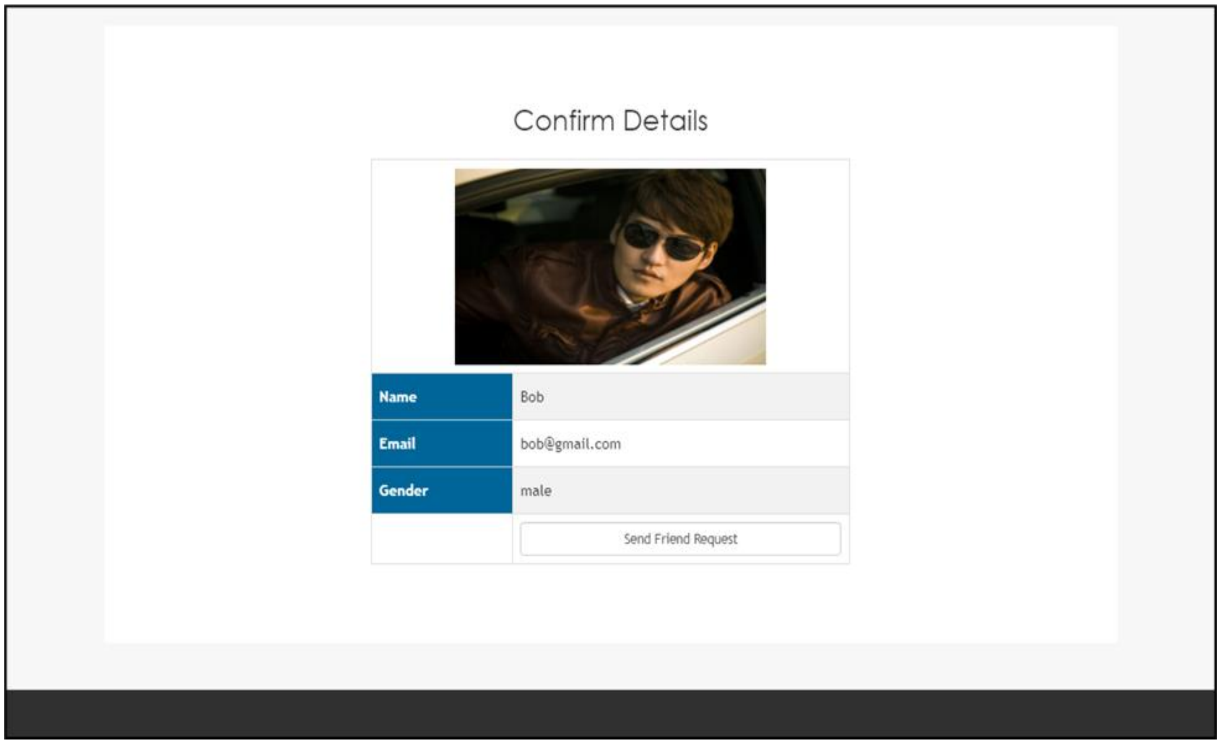


Figure 6.4: Search Friends option



Name	Bob
Email	bob@gmail.com
Gender	male

Figure 6.5: Send Friend request

View Friend Requests and Decision on requests

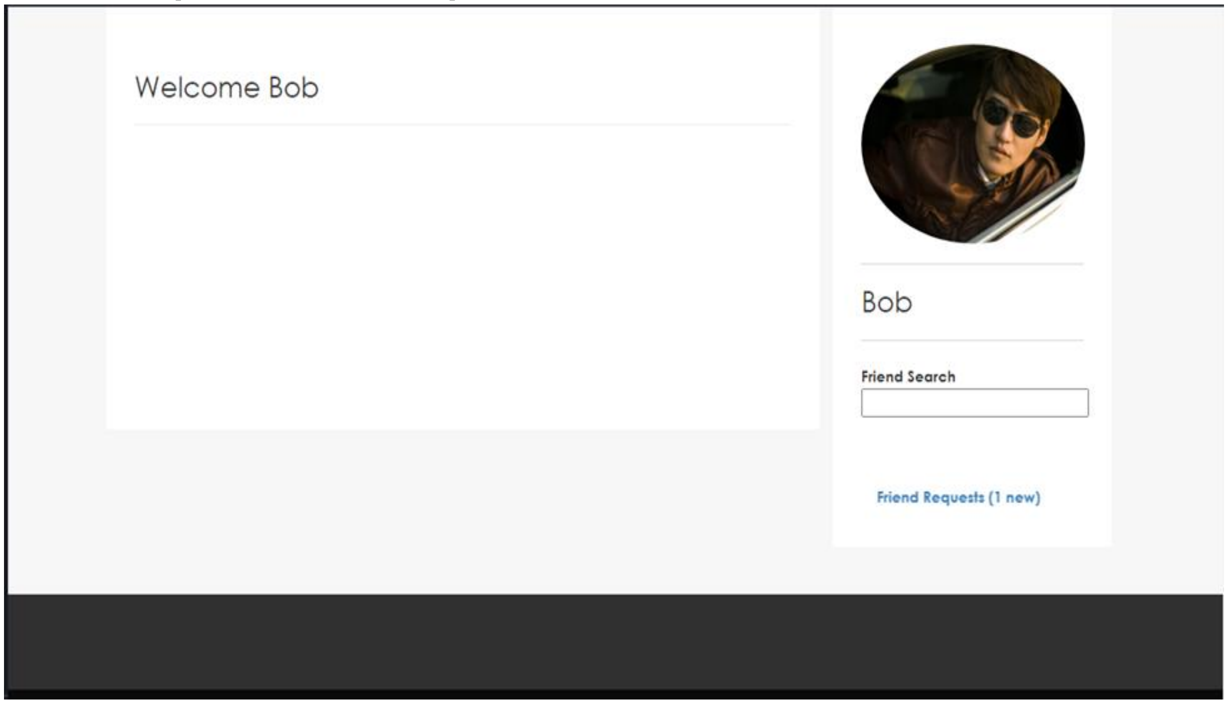


Figure 6.6: View Friend request

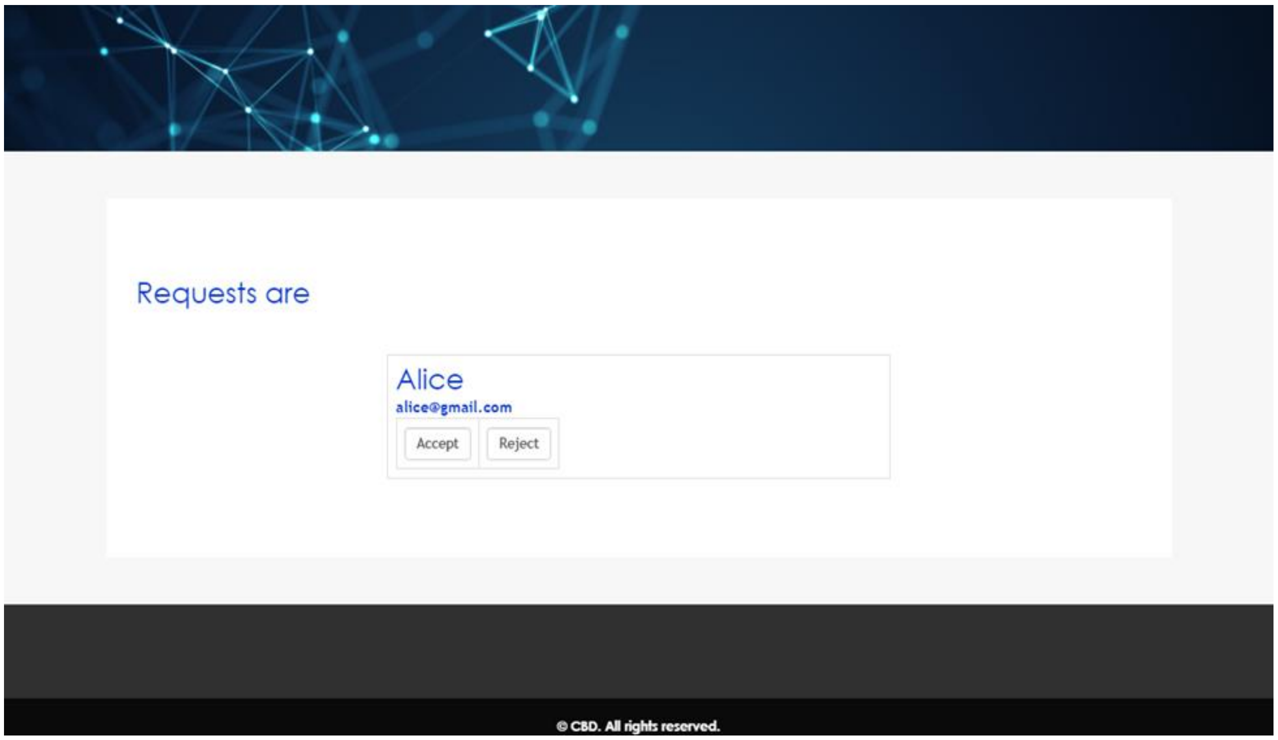


Figure 6.7: Decision on friend request

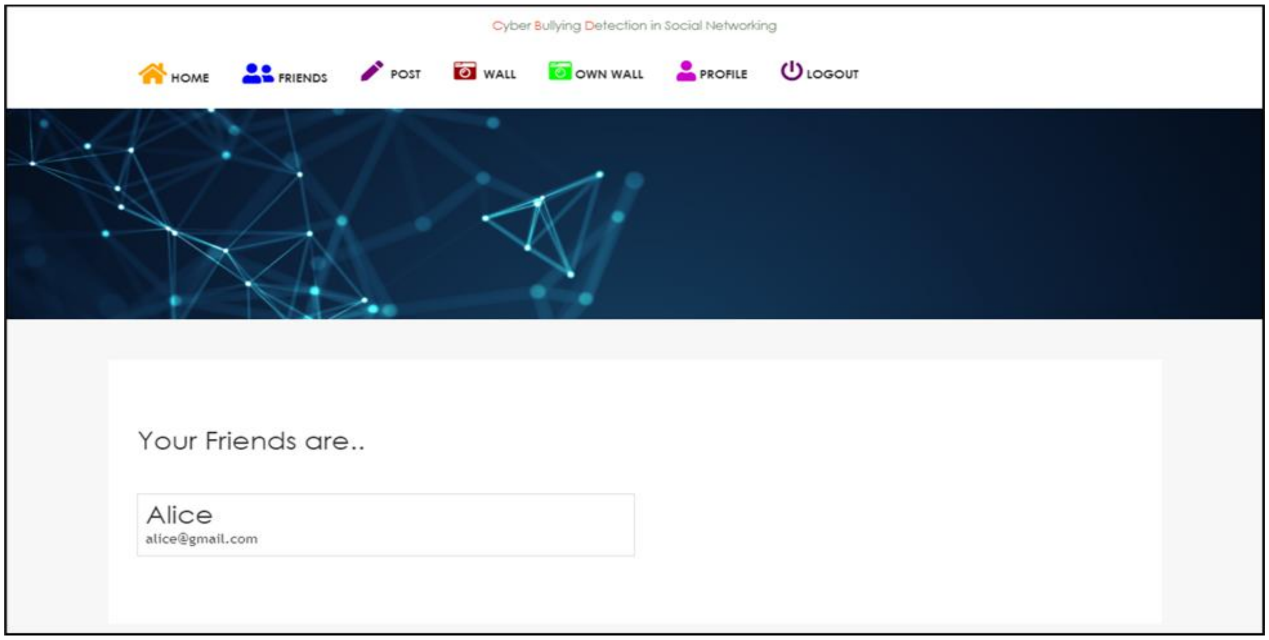


Figure 6.8: Decision on friends

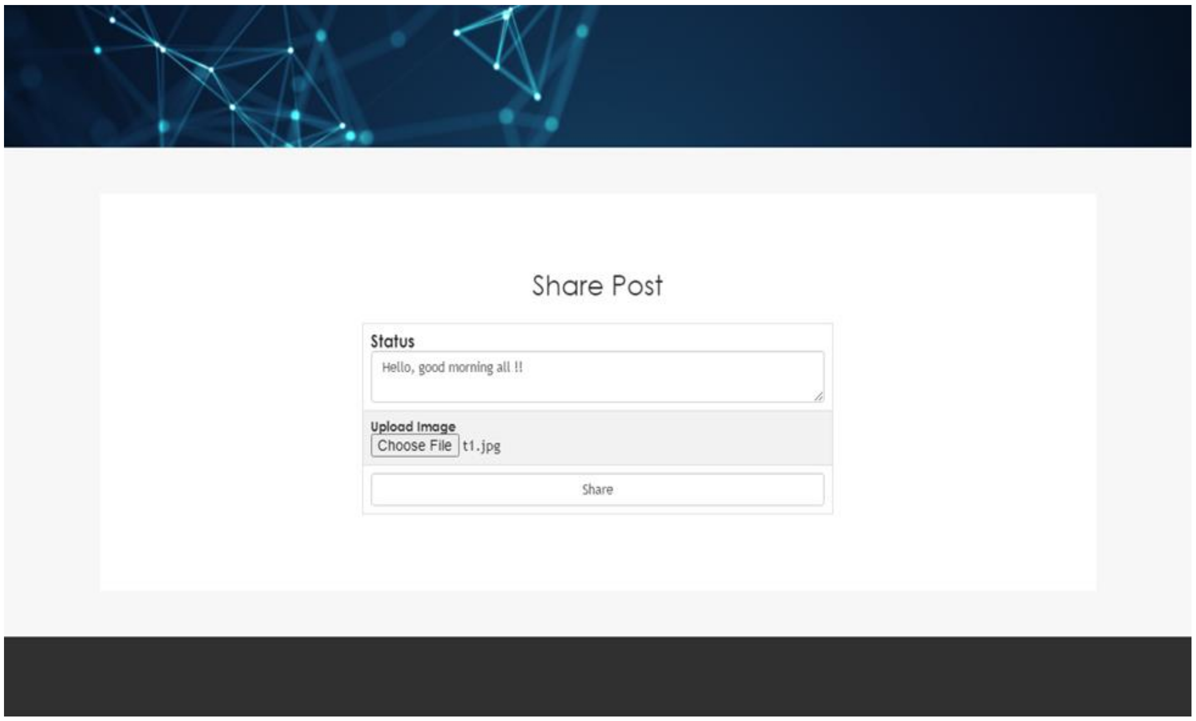


Figure 6.9: Post Share

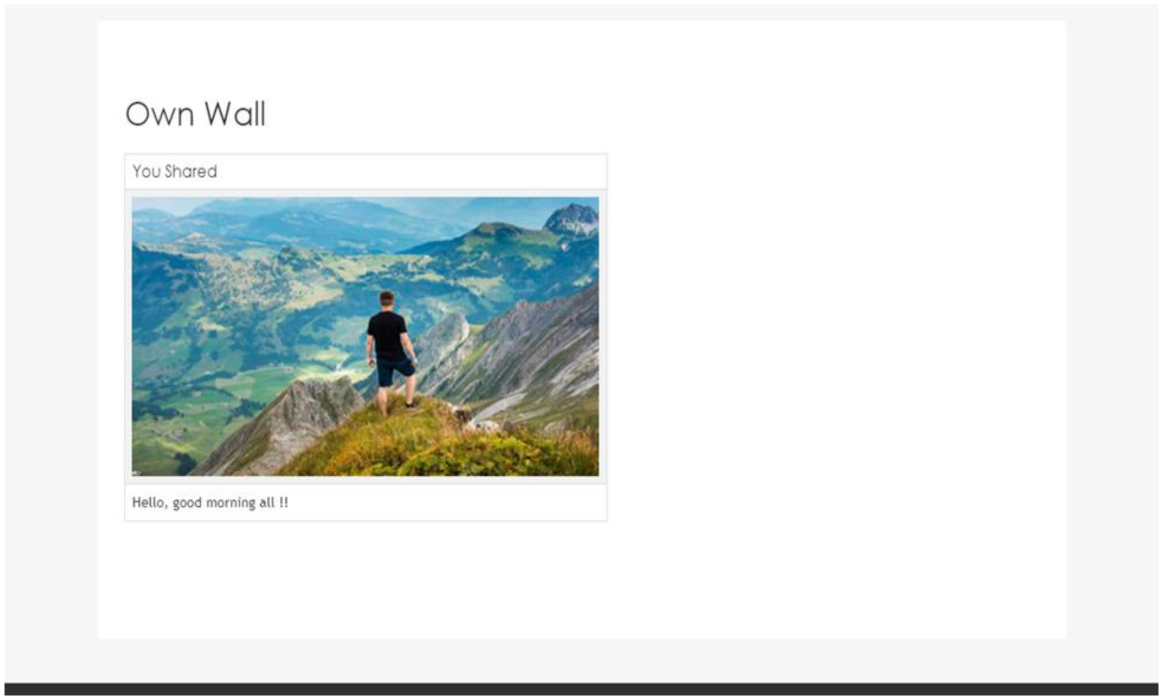


Figure 6.10: View Own Wall

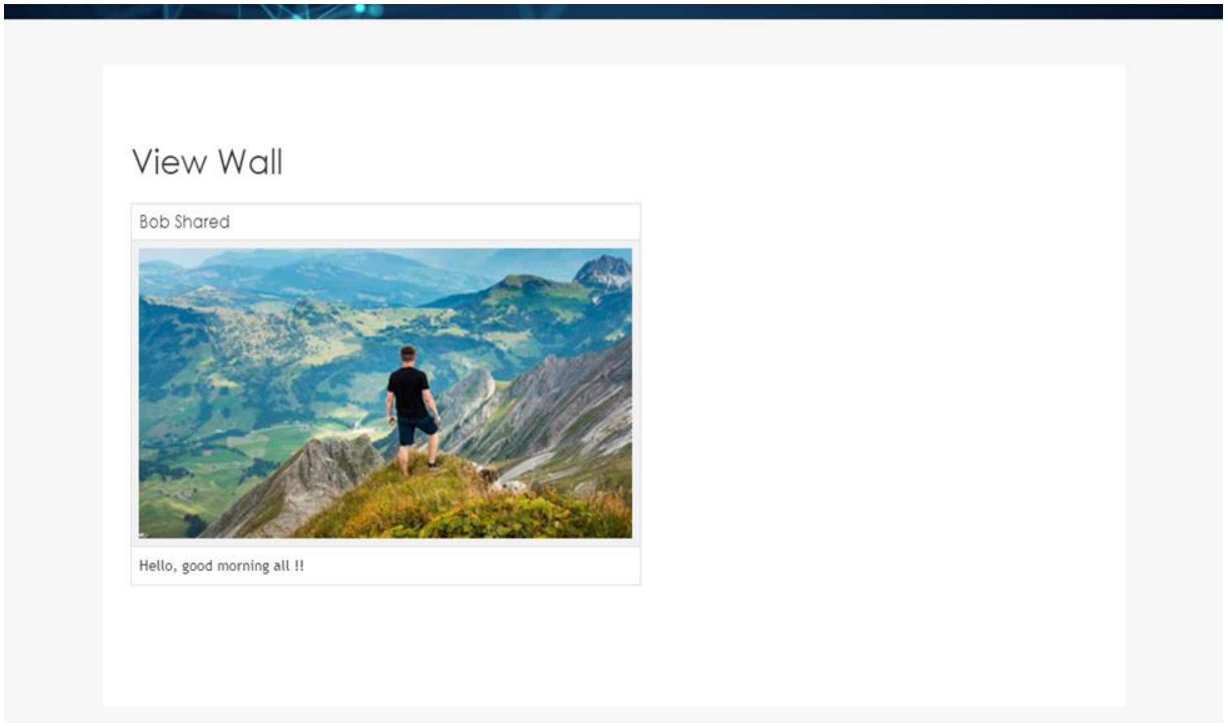


Figure 6.11: View Wall

5.CONCLUSION

In current trend many social networking sites are created and provide services of communications, multi-media services, e-commerce etc immensely. Lot of anonymous user accounts are being created very rapidly. We need to focus on tracking the anonymous users. In our project we have calculated the user behavior according to the chat statements of the user which he/she does with others. By taking advantage of Machine Learning algorithms we classify the anonymous users. Here we are using Naïve Bayes algorithm to perform the classification of the users.

6.ACKNOWLEDEMENTS

We express our gratitude to CMR Technical Campus for their assistance in our project titled "Detection of cyberbullying on social media using machine learning". We extend our sincere thanks to the Chairman, Director, Deans, Head of the Department, Department of Computer Science and Engineering, our guide, and the teaching and non-teaching faculty members for their invaluable suggestions and guidance throughout our work.

REFERENCES

[1] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, “Identifying users across social tagging systems,” in Proc. 5th Int. AAAI Conf. Weblogs Social Media, 2011, pp. 522–525.
[2] Identifying Users Across Social Tagging Systems Tereza Iofciu, Péter Fankhauser, +1 author Kerstin Bischoff, prieto et al. 2009
[3] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manila, “How unique and traceable are usernames?” in Proc. 11th Int. Conf. Privacy Enhancing Technology., 2011, pp. 1–17.
[4] J. Liu, F. Zhang, X. Song, Y. I. Song, C. Y. Lin, and H. W. Hon, “What’s in a name?: An unsupervised approach to link users across communities,” in Proc. 6th ACM Int. Conf. Web Search Data Mining, 2013, pp. 495–504.

10.CERTIFICATION

Certificate of Publication

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY

The Board of
International Journal Of Innovative Research In Technology
is hereby awarding this certificate

SHANMUKANJALI

In recognition of the Publication of the paper entitled

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING SOCIAL MEDIA

Publication In e-Journal

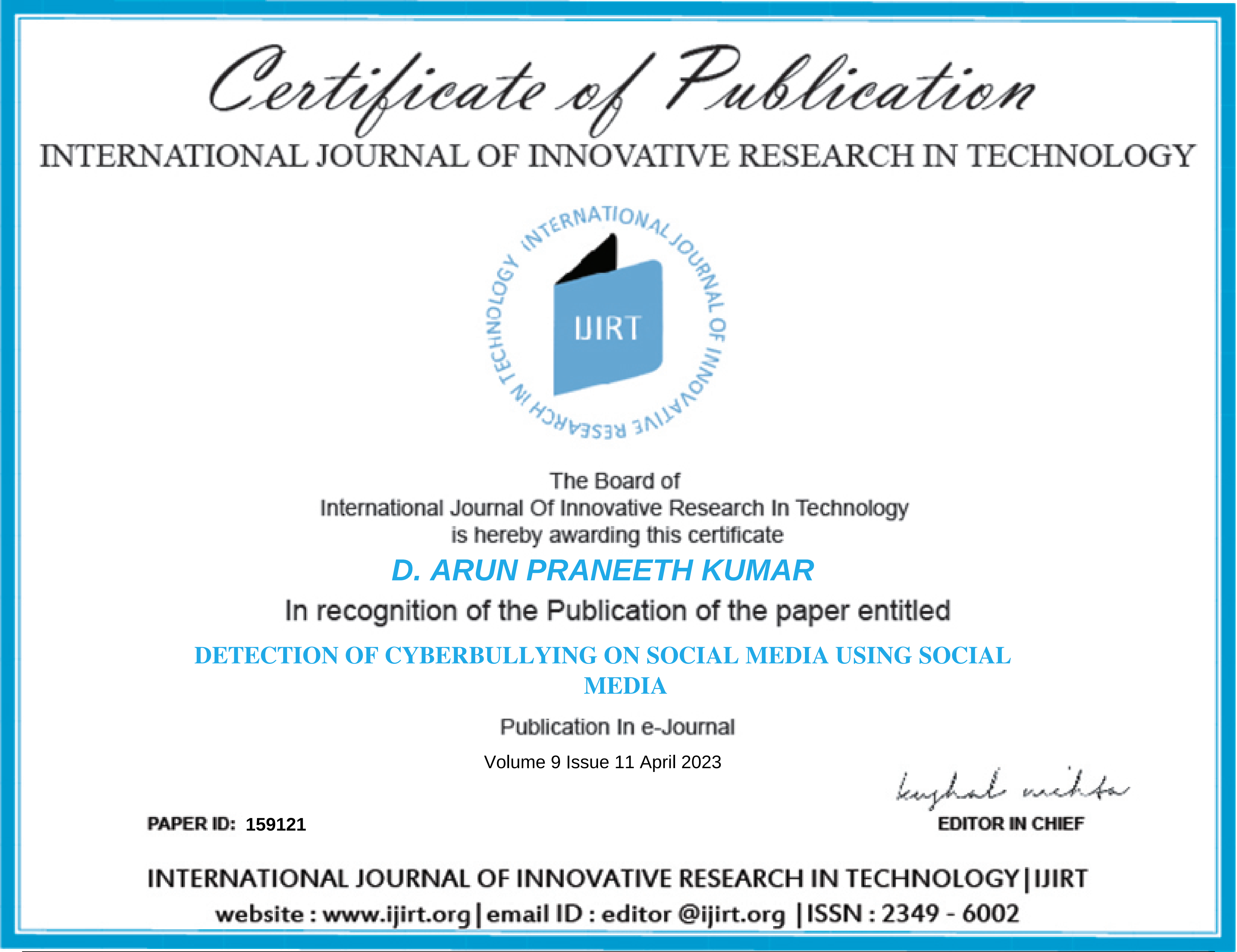
Volume 9 Issue 11 April 2023

PAPER ID: 159121

[illegible]

EDITOR IN CHIEF

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY|IJIRT
website : www.ijirt.org|email ID : editor @ijirt.org |ISSN : 2349 - 6002



Certificate of Publication

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY

The Board of
International Journal Of Innovative Research In Technology
is hereby awarding this certificate

D. ARUN PRANEETH KUMAR

In recognition of the Publication of the paper entitled

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING SOCIAL MEDIA

Publication In e-Journal

Volume 9 Issue 11 April 2023

PAPER ID: 159121

[illegible]
EDITOR IN CHIEF

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY|IJIRT
website : www.ijirt.org|email ID : editor @ijirt.org |ISSN : 2349 - 6002

Certificate of Publication

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY

The Board of
International Journal Of Innovative Research In Technology
is hereby awarding this certificate

P. VARSHEEK REDDY

In recognition of the Publication of the paper entitled

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING SOCIAL MEDIA

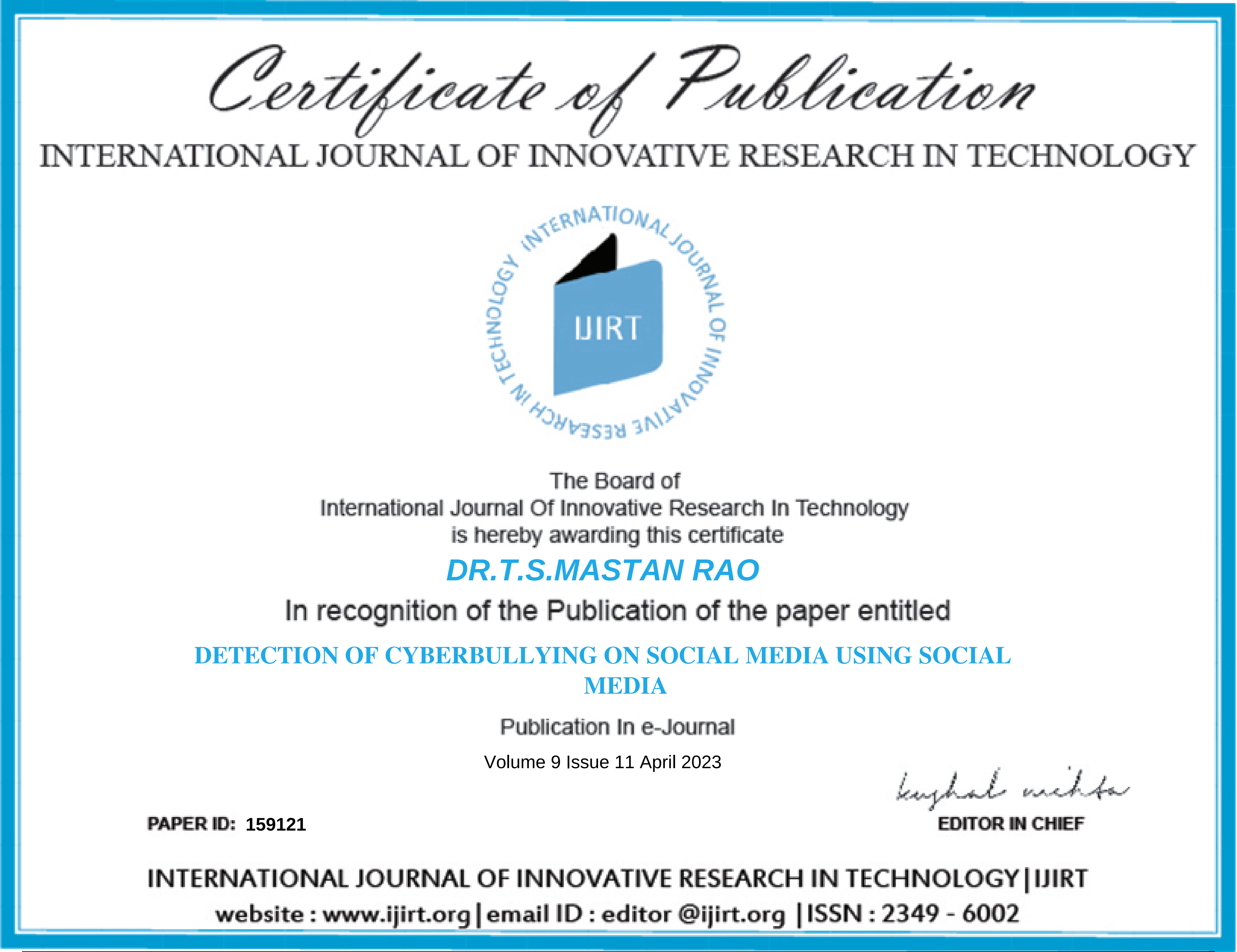
Publication In e-Journal

Volume 9 Issue 11 April 2023

PAPER ID: 159121

[illegible]
EDITOR IN CHIEF

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY | IJIRT
website : www.ijirt.org | email ID : editor @ijirt.org | ISSN : 2349 - 6002



Certificate of Publication

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY

The Board of
International Journal Of Innovative Research In Technology
is hereby awarding this certificate

DR.T.S.MASTAN RAO

In recognition of the Publication of the paper entitled

DETECTION OF CYBERBULLYING ON SOCIAL MEDIA USING SOCIAL MEDIA

Publication In e-Journal

Volume 9 Issue 11 April 2023

PAPER ID: 159121

[illegible]
EDITOR IN CHIEF

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY | IJIRT
website : www.ijirt.org | email ID : editor @ijirt.org | ISSN : 2349 - 6002