# Detecting Child Predators Cyber Harassers On Social Media

A Thesis

Submitted in partial fulfilment of the Requirements for the award of the Degree of **MASTER OF COMPUTER APPLICATIONS**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**MAVILLA MEGHAMALA**

### (19001F0054)

Under the esteemed guidance of

**Dr. K. MADHAVI M.Tech, Ph.D.**

**Associate professor**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR COLLEGE OF ENGINEERING (*Autonomous*)**

**ANANTHAPURAMU-515002 ANDHRA PRADESH-INDIA 2022**

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR COLLEGE OF ENGINEERING (*Autonomous*)**

**ANANTHAPURAMU-515002**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled “**Detecting Child Predators Cyber Harassers On Social Media**” is bonafide work of **Ms. MAVILLA MEGHAMALA**, bearing Admission number **19001F0054**, submitted to the faculty of Computer Science & Engineering, in partial fulfillment of the requirements for the award of degree of **MASTER OF COMPUTER APPLICATIONS** from Jawaharlal Nehru Technological University Anantapur, College of Engineering (Autonomous), Ananthapuramu.

Signature of the Supervisor Signature of Head of the Dept.

**Dr. K. MADHAVI M. Tech, Ph.D.. Dr. K. MADHAVI M. Tech, Ph.D.**

Associate professor Associate professor & HOD

Dept. of Computer Science & Eng., Dept. of Computer Science & Eng.,

JNTUA College of Engineering JNTUA College of Engineering

Ananthapuramu-515002. Ananthapuramu-515002.

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR COLLEGE OF ENGINEERING (*Autonomous*)**

**ANANTHAPURAMU-515002**

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



**DECLARATION**

I am **MAVILLA MEGHAMALA**, bearing admission no. **19001F0054**, hereby declare that the project report entitled **“Detecting Child Predators Cyber Harassers On Social Media”** under the esteemed guidance of **Dr. K. MADHAVI M. Tech, Ph.D.** Associate professor, JNTUA Ananthapuramu is submitted in partial fulfillment of the requirements for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** in Computer Science & Engineering.

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

**MAVILLA MEGHAMALA (19001F0054)**

# Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my guidance for all of them.

The person I would like to thank is my guide **Dr. K. MADHAVI M. Tech, Ph.D.,** Associate Professor of Computer Science and Engineering Department, JNTUA College of Engineering (Autonomous), Ananthapuramu. Her wide knowledge and logical way of thinking have made a deep impression on me. Her understanding, encouragement and personal guidance have provided the basis for this thesis. Her source of inspiration for innovative ideas and his kind support is well to all her students and colleagues.

I wish to thank **Dr. K. MADHAVI M. Tech, Ph.D.,** Associate Professor and Head of Computer Science and Engineering Department, JNTUA College of Engineering (Autonomous), Ananthapuramu. Her wide support, knowledge and enthusiastic encouragement have impressed me to better involvement into my project thesis and technical design also her ethical morals helped me to develop my personal and technical skills to deploy my project in success.

I wish to thank **Prof. B. DURGA PRASAD,** Vice-Principal of JNTUA College of Engineering (Autonomous), Ananthapuramu who has extended his support for the success of this project.

I wish to thank **Prof. P. SUJATHA,** Principal of JNTUA College of Engineering (Autonomous), Ananthapuramu who has extended her support for the success of this project.. I wish to thank teaching staff and non-teaching staff of Computer Science and Engineering Department, JNTUA College of Engineering (Autonomous) Ananthapuramu, for

their support to complete my project successfully.

**MAVILLA MEGHAMALA (19001F0054)**

# Abstract

Online predators try to gradually seduce their targets through attention, affection, kindness, and even gifts, and often devote considerable time, money and energy to this effort. They are aware of the latest music and hobbies likely to interest kids. They listen to and sympathize with kids' problems. They also try to ease young people's inhibitions by gradually introducing sexual content into their conversations or by showing them sexually explicit material. The solution will detect suspect profiles based on child grooming behavior patterns followers, hate speech provokers, stalking and bullying mentality profiles and explicit content explorers ( postings, comments) on social media platforms and other websites.

The existing system, there exists various child predator detection system which are used in gaming, audio chat and in various online entertainment platform. While playing games or for using online audio chat there exists a child predator system which detects an online sexual harassment and prevent child from getting abused or getting harassed by sexual predator as this existing system is only used when the children are playing games on internet or doing any audio chats. As now we are in internet era various children are now days using social media platform for various social activities. They are mostly active on social media so to prevent child harassment we need a child predator detection system for social media.

The proposed system has to using Internet communication tools such as chat rooms, e-mail, and instant messaging can put children at potential risk of encountering online predators. The anonymity of the Internet means that trust and intimacy can develop quickly online. Predators take advantage of this anonymity to build online relationships with inexperienced young people. The solution will detect suspect profiles based on child grooming behavior patterns followers, hate speech provokers, stalking and bullying mentality profiles and explicit content explorers ( postings, comments) on social media platforms and other websites.

# Contents

|  |  |
| --- | --- |
| **Acknowledgement** | **Iv** |
| **Abstract** | **V** |
| **List of figures** | **Viii** |
| **Chapters** | **Page No** |
| **1. Introduction** | **1** |
| 1.1 Objective | 1 |
| 1.2 Existing System & Disadvantages | 1 |
| 1.3 Proposed System & Advantages | 2 |
| 1.4 Architecture | 3 |
| 1.5 Modules | 4 |
| 1.6 Organization of Thesis | 4 |
| **2. System Analysis** | **5** |
| 2.1 Process Model Used With Justification | 5 |
| 2.2 Feasibility Study | 7 |
| 2.2.1 Economical Feasibility | 7 |
| 2.2.2 Technical Feasibility | 7 |
| 2.2.3 Social Feasibility | 8 |
| 2.3 Software Requirement Specification | 8 |
| * 1. Software Specifications   2. Hardware Specifications | 9  9 |
| **3. System Design** | **10** |
| 3.1 Input Design | 10 |
| 3.2 Output Design | 11 |
| 3.3 UML Diagrams | 12 |
| 3.3.1 Use Case Diagram | 13 |
| 3.3.2 Class Diagram | 14 |
| 3.3.3 Sequence Diagram | 15 |
| 3.3.4 Collaboration Diagram | 16 |
| 3.3.5 Activity Diagram | 17 |
| 3.3.6 Component Diagram | 18 |

1. Testing Strategies 19
   1. Testing in strategies 20
      1. [White box Testing 20](#_bookmark0)
      2. [Black box Testing 20](#_bookmark1)
   2. [Test Results 21](#_bookmark2)
2. Screenshots 22
3. Conclusion 31

[References 32](#_bookmark3)

Appendix 33

|  |  |
| --- | --- |
| **List of Figures** | |
| **Figure Name**  1.1 Architecture | **Page No**  3 |
| 3.3.1 Use Case Diagram | 13 |
| 3.3.2 Class Diagram | 14 |
| 3.3.3 Sequence Diagram | 15 |
| 3.3.4 Collaboration Diagram | 16 |
| 3.3.5 Activity Diagram | 17 |
| 3.3.6 Component Diagram | 18 |
| 5.1 Home Page | 22 |
| 5.2 Admin Login Page | 23 |
| 5.3 Add Category | 24 |
| 5.4 Add Words | 25 |
| 5.5 User Registration | 26 |
| 5.6 User Login | 27 |
| 5.7 Post Content | 28 |
| 5.8 View All Posts Screen | 29 |
| 5.9 View Cyber Harassers | 30 |

viii

**Chapter 1**

# Introduction

Child predator detection system on social media is a web based application. This project aims to detect child predator comments and post on social media like fb, insta etc and send report to cyber cell admin. To develop an well-designed database to store all comments and post of social online contact of children in pedophiles is a rapidly growing problem on social media. As of march 2014, the national society for the hindrance of cruelty to kids (nspcc), reported that i) 12-tone system of 11-16 year olds within the kingdom have received unwanted sexual messages; and ii) 8% of 11-16 year olds in the uk have received requests to send or respond to a sexual message. The detection of kids cyber sexual-offenders is so a crucial issue that must be addressed. Kids in their teens have began to use social media as their main means that of communication. Moreover, a recent study of cognition, adolescents and mobile phones (scamp) has revealed that 70% of 11-12 year olds in the uk now own a mobile phone rising to 90% by age 14. A common attack of pedophiles is the so-called online child grooming, xmedia outlets. Such grooming consists of building a trust-relationship with a minor, which finally leads into convincing a child to meet them in person. Previous research on detecting cyber pedophilia online, including the efforts of the first international sexual predator identification competition.

## Objective

The main objective of our project is “To detect child predator base on comments and post of social media account and send predator record to cyber cell admin”.

## Existing Method

There exists various child predator detection system which are used in gaming, audio chat and in various online entertainment platform. While playing games or for using online audio chat there exists a child predator system which detects an online sexual harassment and prevent child from getting abused or getting harassed by sexual predator as this existing system is only used when the children are playing games on internet or doing any audio chats. As now we are in internet era various children are now days using social media platform for various social activities. They are mostly active on social media so to prevent child harassment we need

a child predator detection system for social media.

### Disadvantages

* + - Message-Based Detection**.**
    - Author-Based Detection**.**
    - Conversation-Based Detection**.**

## Proposed System

The proposed system has to using Internet communication tools such as chat rooms, e-mail, and instant messaging can put children at potential risk of encountering online predators. The anonymity of the Internet means that trust and intimacy can develop quickly online. Predators take advantage of this anonymity to build online relationships with inexperienced young people. The solution will detect suspect profiles based on child grooming behavior patterns followers, hate speech provokers, stalking and bullying mentality profiles and explicit content explorers ( postings, comments) on social media platforms and other websites.

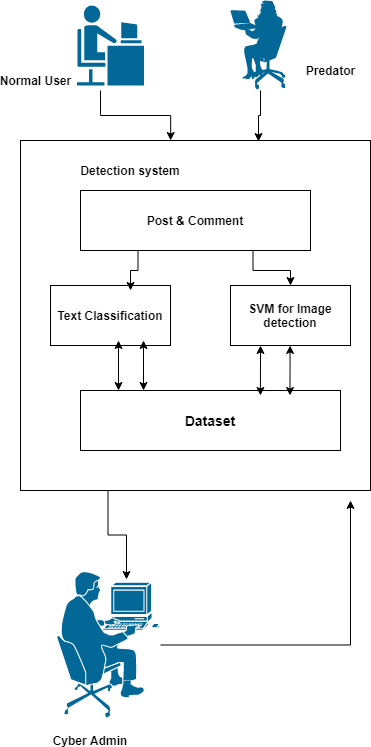
## Advantages

* + - Text Detection
    - Image detection in our system result base on text classification and image processing.

## Architecture

The figure 1.1 represents mainly flow of Child predator detection system on

social media.



**Fig 1.1** Architecture of Child predator detection system on social media.

## Modules

In this system consists of following modules. They are

### User Module

* **Admin Module User Module**

User has to register with necessary details along with details. User can able to login with the username, password along with images whatever they are provided at registration. Then the user can login with security images. He/ She can able to login with security then they can able to view their profile and view account balance.

### Admin Module

Admin can login with username and password. Admin can able activate users. admin can also able to view user details.

## 1.5 Organization of Thesis

The rest of the Thesis is organized as follows.

### Chapter 2:

This chapter explains about methodology & Algorithms, feasibility study and software requirement specifications.

### Chapter 3:

This chapter explains the entire system design of the project which explains how to develop the project and UML diagrams.

### Chapter 4:

It explains the software testing strategies of the project.

### Chapter 5:

This chapter covers the screenshots of the proposed system.

### Chapter 6:

Conclusion about the project and explains about the feature expansions.

# Chapter 2 System Analysis

System analysis is the process of observing systems for troubleshooting or development purposes. It is applied to information technology, where computer-based systems require defined analysis according to their makeup and design.

## Process Model Used With Justification

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

### Stages in SDLC:

* + - Requirement Gathering
    - Analysis
    - Designing
    - Coding
    - Testing
    - Maintenance

### Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high- level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

### Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

### Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

### Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

## Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

### Maintenance

Outer rectangle represents maintenance of a project, maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will under go training on that particular assigned category.

For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

## Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* + 1. Economical Feasibility
    2. Technical Feasibility
    3. Social Feasibility.

### Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### Social Feasibility

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

## Software Requirement Specification:

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

### Functional and non-functional requirements

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

### Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non- functional requirements.

Examples of functional requirements:

* Authentication of user whenever he/she logs into the system.
* System shutdown in case of a cyber-attack.
* A verification email is sent to user whenever he/she register for the first time on some software system.

### Non-functional requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavior requirements. They basically deal with issues like:

* + Portability
  + Security
  + Maintainability
  + Reliability
  + Scalability
  + Performance
  + Reusability
  + Flexibility

Examples of non-functional requirements:

* + Emails should be sent with a latency of no greater than 12 hours from such an activity.
  + The processing of each request should be done within 10 seconds.
  + The site should load in 3 seconds whenever of simultaneous users are > 10000.

## Software Specifications:

* Operating system : Windows 7
* Scripting Language : Java
* Database : MYSQL
* IDE : Net beans

## Hardware Specifications:

* Processor : I3/Intel Processor or above
* RAM : 1GB
* Hard Disk : 120GB

# Chapter 3

**System Design**

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

## Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things:

* + - What data should be given as input?
    - How the data should be arranged or coded?

### Objectives

* + - Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
    - It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
    - When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

## Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intel agent output design improves the system’s relationship to help user decision-making.

* + - Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
    - Select methods for presenting information.
    - Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* + - Convey information about past activities, current status or projections of the Future.
    - Signal important events, opportunities, problems, or warnings.
    - Trigger an action.

## UML Diagrams

UML represents Unified Modeling Language. UML is an institutionalized universally useful showing dialect in the subject of article situated programming designing. The fashionable is overseen, and become made by way of, the Object Management Group.

The goal is for UML to become a regular dialect for making fashions of item arranged PC programming. In its gift frame UML is contained two noteworthy components: A Meta show and documentation. Later on, a few types of method or system can also likewise be brought to; or related with, UML.

The Unified Modeling Language is a popular dialect for indicating, Visualization, Constructing and archiving the curios of programming framework, and for business demonstrating and different non-programming frameworks.

### Goals:

The Primary goals inside the plan of the UML are as in step with the subsequent:

* + - Provide clients a prepared to-utilize, expressive visual showing Language on the way to create and change massive models.
    - Provide extendibility and specialization units to make bigger the middle ideas.
    - Be free of specific programming dialects and advancement manner
    - Provide a proper cause for understanding the displaying dialect.
    - Encourage the improvement of OO gadgets exhibit.
    - Support large amount advancement thoughts, for example, joint efforts, systems, examples and components

### Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. The figure 3.3.1 shows the purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Fig 3.3.1 Use case diagram of Child predator detection system on social media

### Class Diagram

Class diagram in Unified Modeling Language (UML) is a static diagram. The figure

* + 1. shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

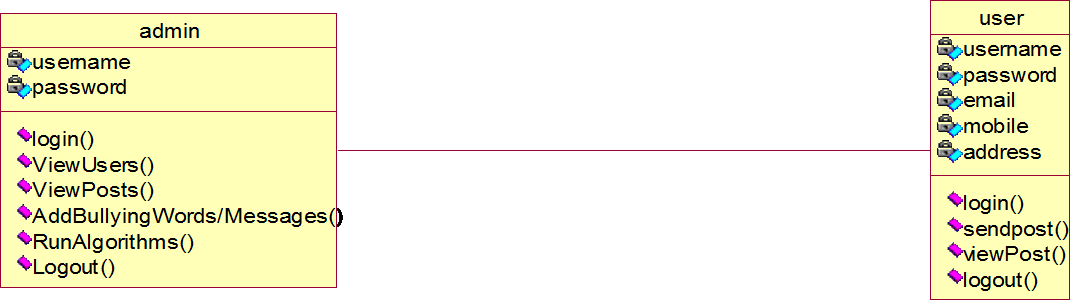


Fig 3.3.2 Class diagram of Child predator detection system on social media.

### Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that the figure 3.3.3 shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagram.

Register

View Users And Activate

Login

Add Bullying Words/Messages

Send Post

View Posts

Run Algorithms

View Post

Logout

Logout

Database

User

Admin

Fig 3.3.3 Sequence diagram of Child predator detection system on social media.

### Collaboration Diagram

In collaboration diagram the method call sequence is indicated by some numbering technique as shown in figure 3.3.4. The number indicates how the methods are called one after another. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram describes the object organization.

9: Logout

Admin

2: View Users And Activate 4: Add Bullying Words/Messages

6: View Posts

7: Run Algorithms

10: Logout

1: Register

User

Databas e

3: Login

5: Send Post

8: View Post

Fig 3.3.4 collaboration diagram of Child predator detection system on social media

## Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

login

View user

login

Monitor post

Add bullying words / message

Run algorithem

yes

no

Fig 3.3.5 Activity diagram of Child predator detection system on social media

## Component diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node

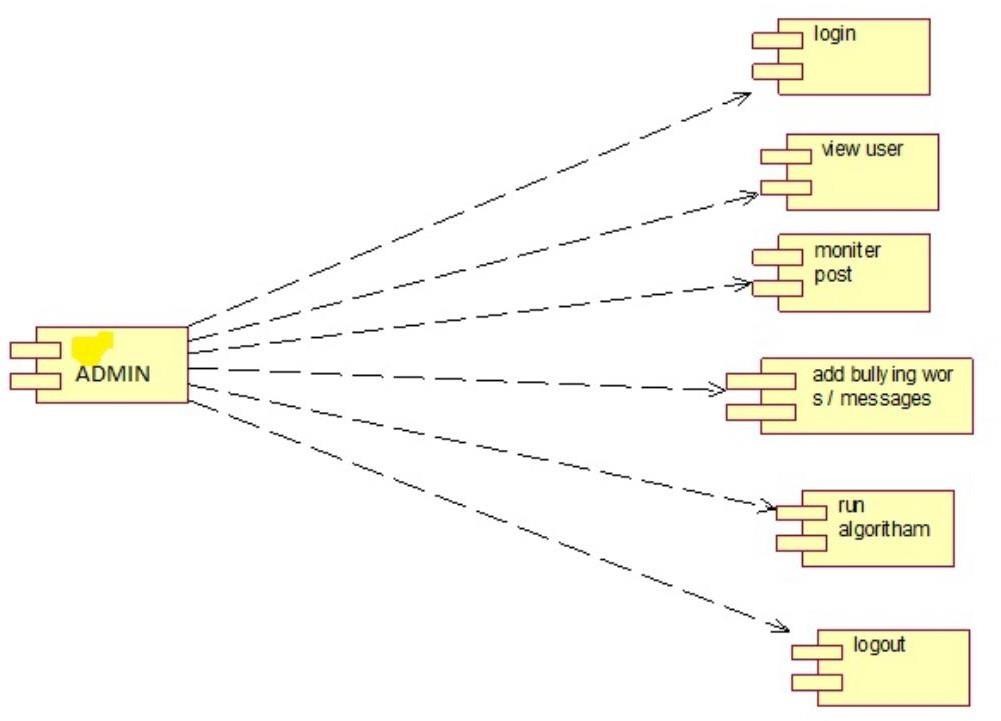


Fig 3.3.6 Component diagram of Child predator detection system on social media

# Chapter 4

**System Testing**

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

## Testing Activities

Different levels of testing are used in the testing process, each level of testing aims to test different aspects of the system. The basic levels are:

## Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

## Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

## Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive and the effective value of current tests is determined.

## System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

## Types of Testing

* + 1. White box or structural testing.
    2. Black box or functional testing

### White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and

responds to outputs without considering how the software works.

## Test Results

### Unit Testing Results

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

### Features to be tested

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications.

e.g. components in a software system or – one step up – software applications at the company level interact without error.

### Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. **Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# Chapter 5

**Screenshots**

The Graphical User Interface helps the user to interact friendly with the application. It enhances the appearance of the application. The GUI magnetizes the attention of the user. The Graphical User Interface involved in the project is shown below.

## Home Page:

Figure 5.1 represents the home page.



Fig 5.1 Home Page of Child Predators Detection

## Admin Login Screen:

Figure 5.2 represents the Admin Login Page. Here the Admin is login to the Web page.

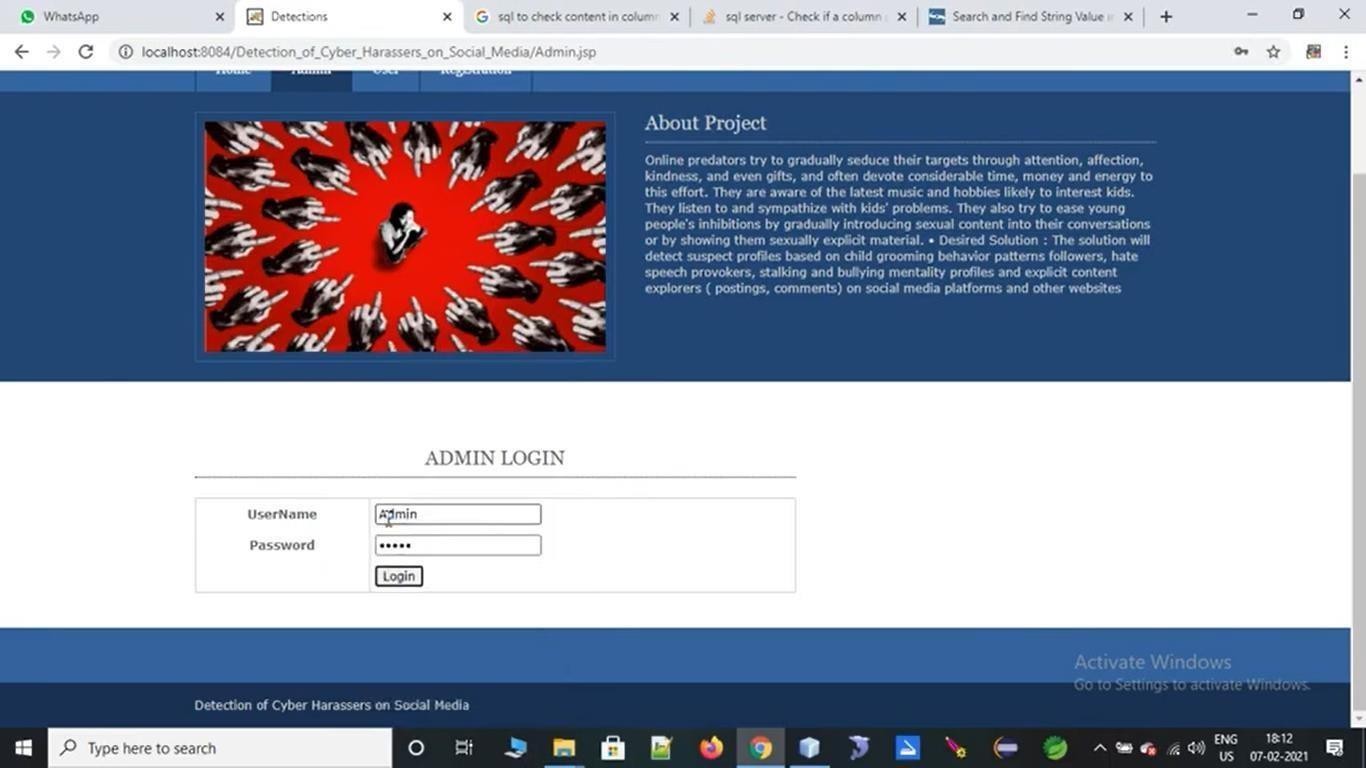


Fig 5.2 Admin Login Page to login

## Add Category:

Figure 5.3 represents the Add Category. Here the Admin add category in the web page.

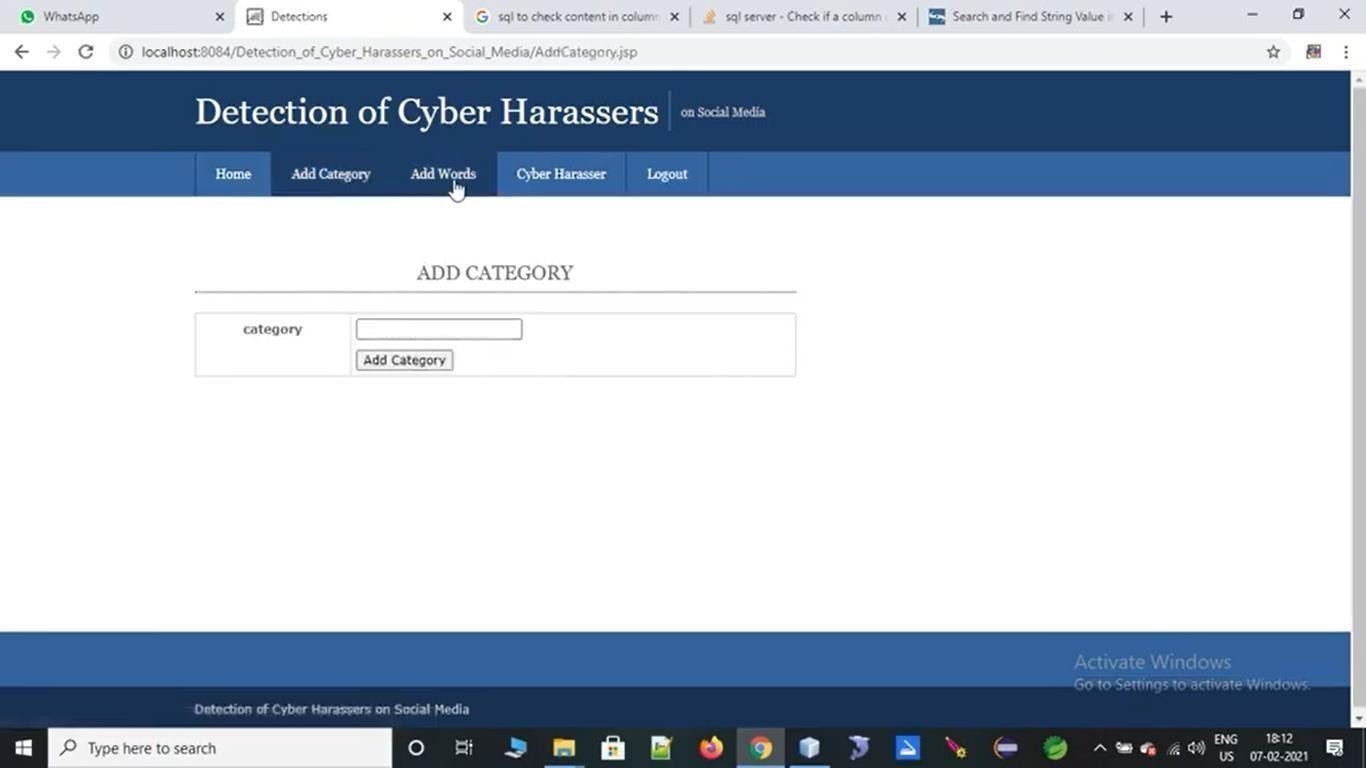


Fig 5.3 Add Category by Admin

## Add Words:

Figure 5.4 represents the Add Words. Here the Admin Add the Words.

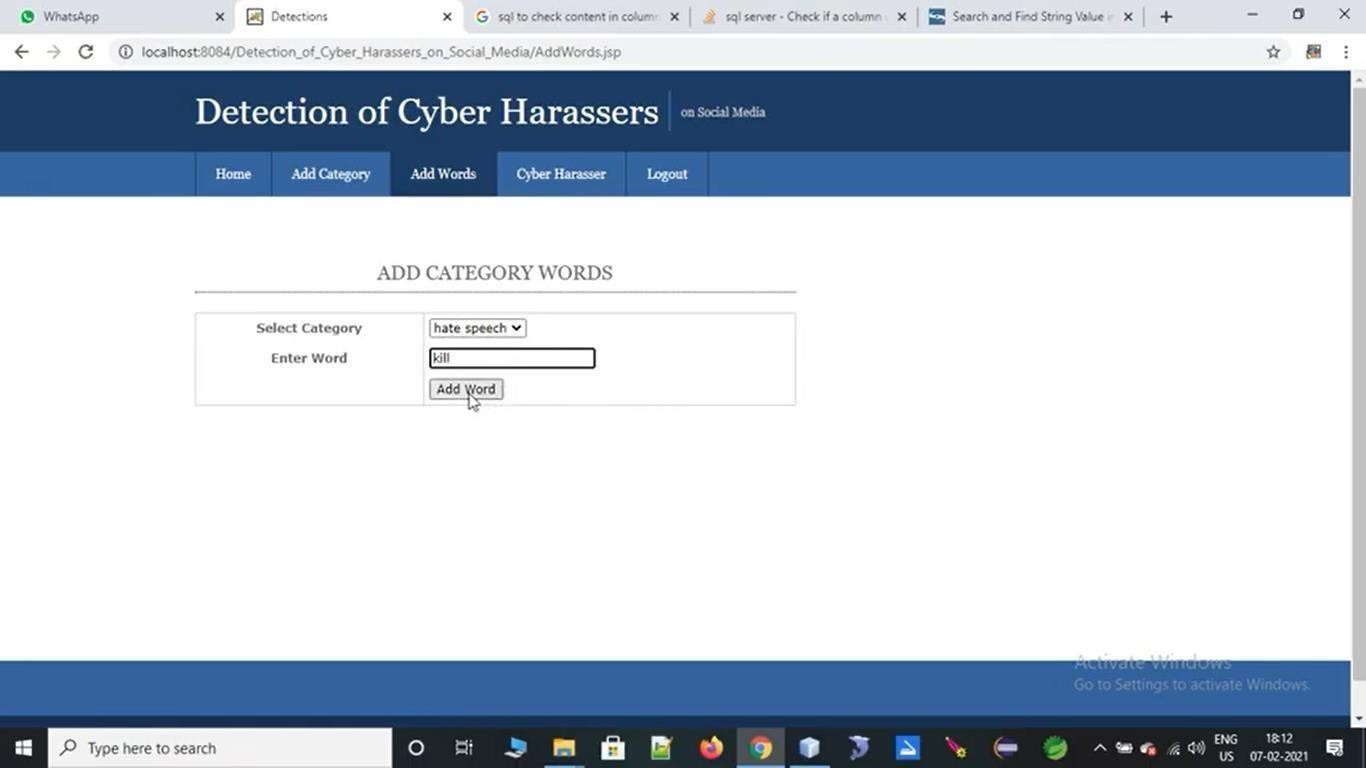


Fig 5.4 Add Words by Admin

## 

## User Registration:

Figure 5.5 represents the User Registration. Here the Admin allows User to Register in the Website.

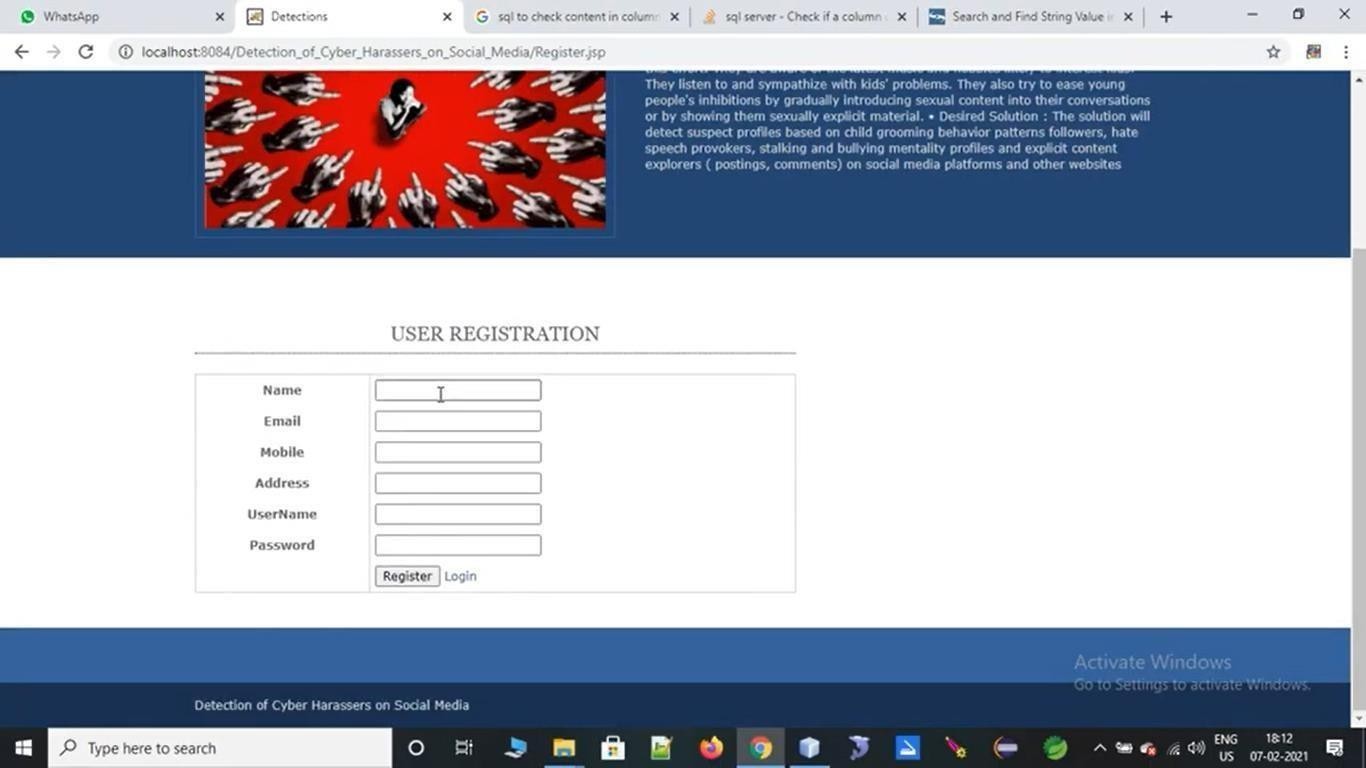


Fig 5.5 User Registration form to the User

## User Login:

Figure 5.6 represents User Login. Here the user able to login who will be already registered.

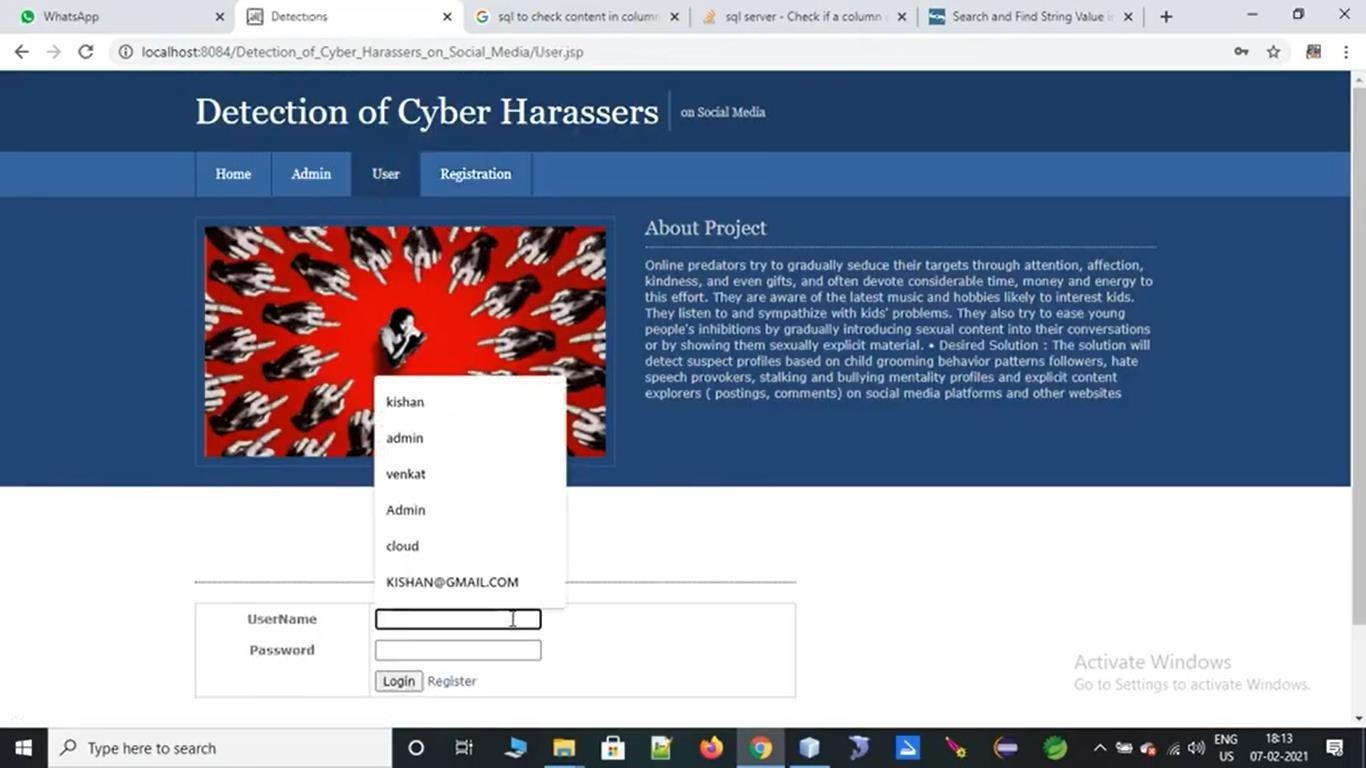


Fig. 5.6 User Login form

## Post Content:

Figure 5.7 represents the Post Content. Here the User Post the Content .

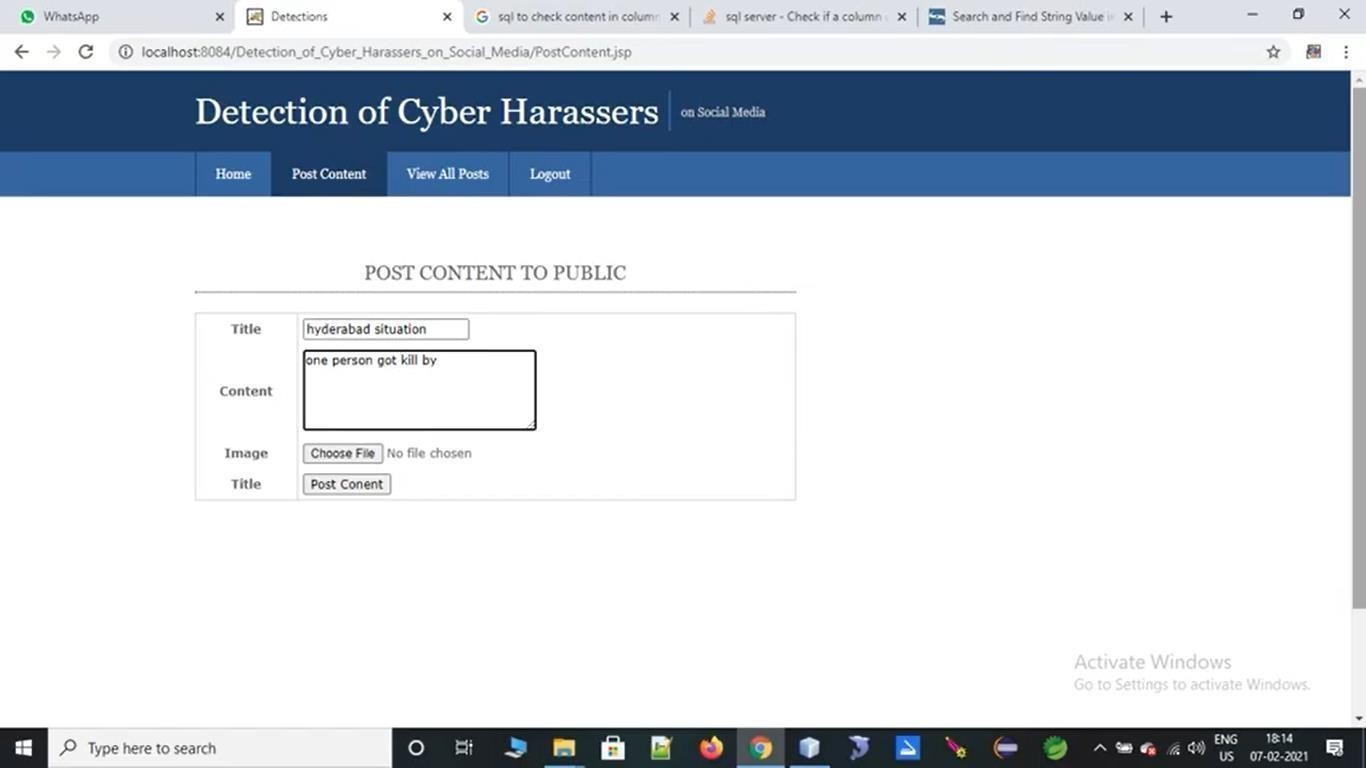


Fig 5.7 Post a Content by User

## View All Posts Screen:

Figure 5.8 represents to View all Posts Screen. Here the Admin View all Posts Screen.

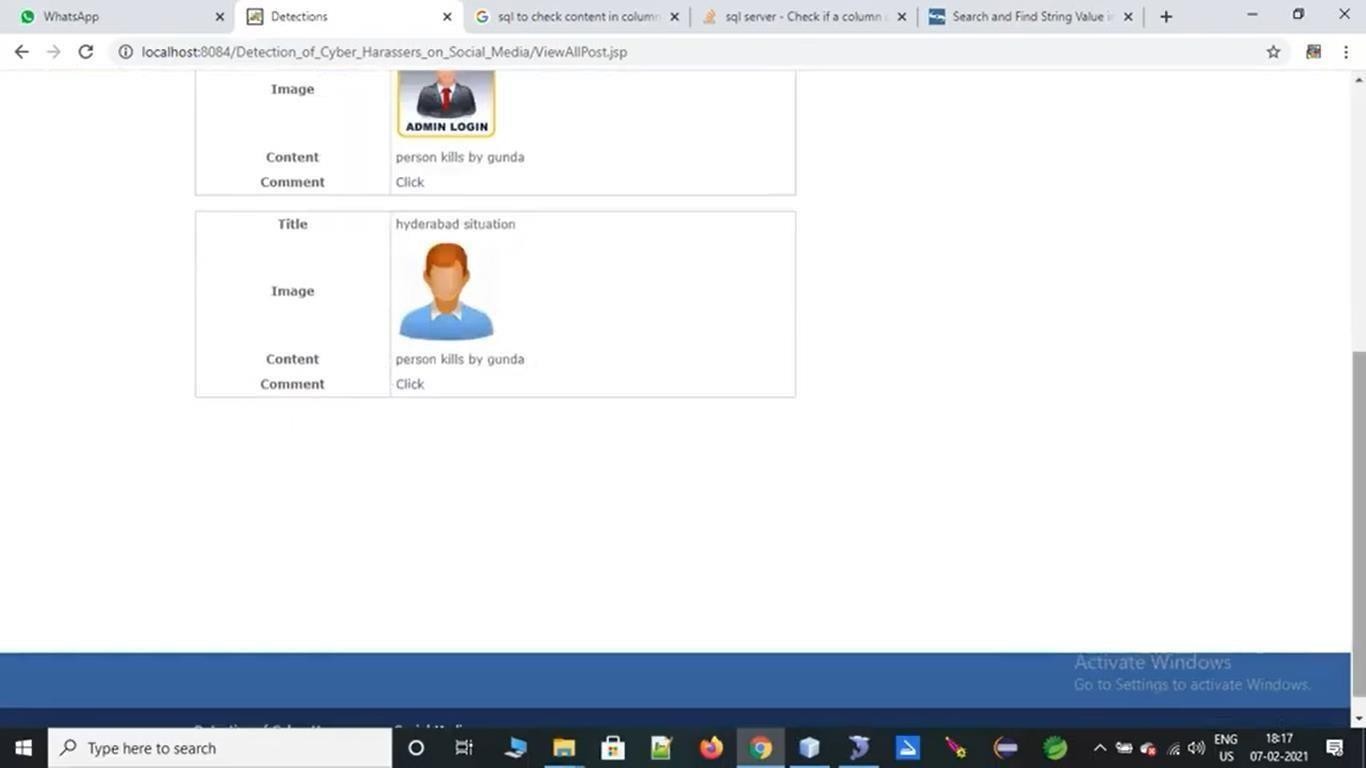


Fig 5.8 View All Posts by Admin

## View Cyber Harassers:

Figure 5.9 represents the view cyber Harassers. Here the admin viewed all Cyber Harassers

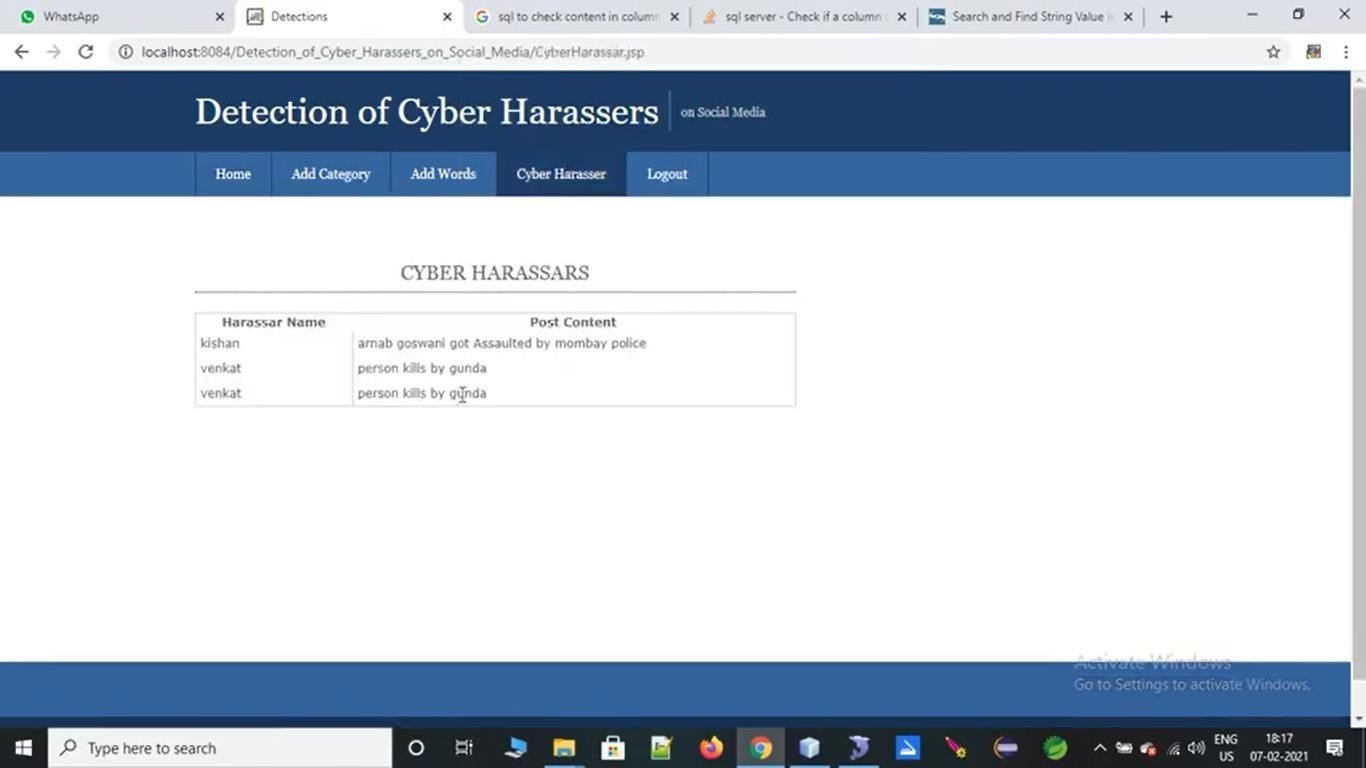


Fig.5.9. View Cyber Harassers by Admin

# Chapter 6 Conclusion

This study reviewed existing literature to detect aggressive behavior on SM websites by using machine learning approaches. We specifically reviewed four aspects of detecting cyber bullying messages by using machine learning approaches, namely, data collection, feature engineering, construction of cyber bullying detection model, and evaluation of constructed cyber bullying detection models. Several types of discriminative features that were used to detect cyber bullying in online social networking sites were also summarized. In addition, the most effective supervised machine learning classifiers for classifying cyber bullying messages in online social networking sites were identified.

One of the main contributions of current paper is the definition of evaluation metrics to successfully identify the significant parameter so the various machine learning algorithms can be evaluated against each other. Most importantly we summarized and identified the important factors for detecting cyber bullying through machine learning techniques specially supervised learning. For this purpose, we have used accuracy, precision recall and f-measure which gives us the area under the curve function for modeling the behaviors in cyber bullying. Finally, the main issues and open research challenges were described and discussed.

# REFERENCES

1. V. Subrahmanian and S. Kumar, ``Predicting human behavior: The next frontiers,'' Science, vol. 355, no. 6324, p. 489, 2017.
2. M. A. Al-Garadi, K. D. Varathan, and S. D. Ravana, ``Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network,'' Comput. Hum. Behav., vol. 63, pp. 433443, Oct. 2016.
3. L. Phillips, C. Dowling, K. Shaffer, N. Hodas, and S. Volkova, ``Using social media to predict the future: A systematic literature review,'' 2017, arXiv:1706.06134. [Online]. Available: <https://arxiv.org/abs/1706.06134>
4. M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, ``Improving cyberbullying detection with user context,'' in Advances in Information Retrieval. Berlin, Germany: Springer, 2013, pp. 693696.
5. Y. Chen, Y. Zhou, S. Zhu, and H. Xu, ``Detecting offensive language in social media to protect adolescent online safety,'' in Proc. Int. Conf. Privacy, Secur., Risk Trust (PASSAT), Sep.2012, pp. 7180.
6. V. S. Chavan and S. S. Shylaja, ``Machine learning approach for detection of cyber- aggressive comments by peers on social media network,'' in Proc. Int. Conf. Adv. Comput., Commun. Inform. (ICACCI), Aug. 2015, pp. 23542358.

# Appendix-I

## Java

Java is a high-level, general-purpose, object-oriented, and secure programming language developed by James Gosling at Sun Microsystems, Inc. in 1991. It is formally known as OAK. In 1995, Sun Micro system changed the name to Java. In 2009, Sun Micro system take over by Oracle Corporation.

## Editions Of Java

Each edition of Java has different capabilities. There are three editions of Java:

* **Java Standard Editions (JSE):** It is used to create programs for a desktop computer.
* **Java Enterprise Edition (JEE):** It is used to create large programs that run on the server and manages heavy traffic and complex transactions.
* **Java Micro Edition (JME):** It is used to develop applications for small devices such as set-top boxes, phone, and appliances.

## Java Platform

Java Platform is a collection of programs. It helps to develop and run a program written in the Java programming language. Java Platform includes an execution engine, a compiler and set of libraries. Java is a platform-independent language.

## Java Features

* **Simple:** Java is a simple language because its syntax is simple, clean, and easy to understand. Complex and ambiguous concepts of C++ are either eliminated or re- implemented in Java. For example, pointer and operator overloading are not used in Java.
* **Object-Oriented:** In Java, everything is in the form of the object. It means it has some data and behavior. A program must have at least one class and object.
* **Robust:** Java makes an effort to check error at run time and compile time. It uses a strong memory management system called garbage collector. Exception handling and garbage collection features make it strong.
* **Secure:** Java is a secure programming language because it has no explicit pointer and programs runs in the virtual machine. Java contains a security manager that defines the access of Java classes.
* **Platform-Independent:** Java provides a guarantee that code writes once and run anywhere. This byte code is platform-independent and can be run on any machine.
* **Portable:** Java Byte code can be carried to any platform. No implementation-dependent features. Everything related to storage is predefined, for example, the size of primitive data types.
* **High Performance:** Java is an interpreted language. Java enables high performance with the use of the Just-In-Time compiler.
* **Distributed:** Java also has networking facilities. It is designed for the distributed environment of the internet because it supports TCP/IP protocol. It can run over the internet. EJB and RMI are used to create a distributed system.
* **Multi-threaded:** Java also supports multi-threading. It means to handle more than one job a time.

## Standard Data Types

Data types specify the different sizes and values that can be stored in the variable.

There are different types of data types in Java:

* boolean data type
* byte data type
* char data type
* short data type
* int data type
* long data type
* float data type
* double data type
* classes
* interfaces
* arrays

## Variables

A variable is a container which holds the value while the Java Program is executed.

A variable is assigned with a data type.

Variable is a name of memory location. There are three types of variables in java: local, instance and static.

A variable is the name of a reserved area allocated in memory. In other words, it is a name of the memory location. It is a combination of "vary + able" which means its value can be changed.

## Types of Variables

There are three types of variables in [Java:](https://www.javatpoint.com/java-tutorial)

* local variable
* instance variable
* static variable

## Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists. A local variable cannot be defined with "static" keyword.

## Instance Variable

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static. It is called an instance variable because its value is instance-specific and is not shared among instances.

## Static Variable

A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

## Libraries

Java is one of the most popular programming languages. Java provides a rich set of libraries, and its standard Java library is a very powerful that contains libraries such as **java.lang**, **java.util**, and **java.math**, etc. Java provides more than thousands of libraries except standard libraries. Some of the most useful and popular libraries are as follows:

1. Java Standard libraries
2. Apache Commons
3. Jackson
4. Maven
5. Google-json
6. Log5j and Slf4j
7. JUnit
8. Google Guava
9. JAXB
10. HTTP Libraries