

**A MAJOR PROJECT REPORT**  
**ON**  
**RECOGNITION OF DIGITAL HARASSEMENT ON WEB-  
BASED SOCIAL CHANNELS**

Submitted to

**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

In partial fulfillment of the requirements for the award of degree of

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**AFSHA JABEEN** [21D41A0506]

**B. SRAVYA** [21D41A0528]

**CH. USHA** [21D41A0545]

**D. NAVITHA** [21D41A0549]

Under the guidance of

**Prof. CH.GVN.PRASAD**

(HOD-CSE)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution under UGC, accredited by NBA, Affiliated to JNTUH)

Sheriguda (V), Ibrahimpatnam (M), Rangareddy Dist – 501 510

(2024-2025)

# **SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution under UGC, Accredited by NBA, Affiliated to JNTUH)

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



### **CERTIFICATE**

Certified that the Major project entitled “**RECOGNITION OF DIGITAL HARASSEMENT ON WEB-BASED SOCIAL CHANNELS**” is a Bonafide work carried out by **AFSHA JABEEN (21D41A0506), B.SRAVYA (21D41A0528), CH. USHA (21D41A0545), D. NAVITHA (21D41A0549)** in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** of SICET, Hyderabad for the academic year 2024-2025. The Project has been approved as it satisfies academic requirements in respect of the work prescribed for **IV YEAR, II-SEMESTER** of **B.TECH** course.

**INTERNAL GUIDE**

**(Prof. CH. GVN. PRASAD)**  
(HOD-CSE)

**HEAD OF THE DEPARTMENT**

**(Prof. CH. GVN. PRASAD)**

**EXTERNAL EXAMINER**

## ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We are thankful to principal **Prof. Dr. G. Suresh**, for giving us the permission to carry out this project and for providing necessary infrastructure and labs.

We are highly indebted to **Prof. CH. GVN. Prasad**, Head of the Department of Computer Science & Engineering, for his constant motivation, valuable guidance, and support throughout the execution of this project. His insights at every stage greatly contributed to the successful completion of our work.

We would like to thank the Teaching & Non-Teaching staff of Department of Computer Science & Engineering for sharing their knowledge with us.

Last but not the least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

AFSHA JABEEN [21D41A0506]

B. SRAVYA [21D41A0528]

CH. USHA [21D41A0545]

D. NAVITHA [21D41A0549]

# **CONTENTS**

<b>S.NO</b>	<b>PAGE.NO</b>
<b>1. INTRODUCTION</b>	
1.1 Introduction To Project .....	1-2
<b>2. LITERATURE SURVEY .....</b>	<b>3-4</b>
<b>3. SYSTEM ANALYSIS</b>	
3.1 Existing System & Its Disadvantage.....	05
3.2 Proposed System & Its Advantages.....	06
<b>4. SYSTEM REQUIREMENTS</b>	
4.1 Functional Requirements.....	07
4.2 Non-Functional Requirements .....	08
<b>5. SYSTEM STUDY</b>	
5.1 Feasibility Study.....	09
5.2 Feasibility Analysis .....	10
<b>6. SYSTEM DESIGN</b>	
6.1 System Architecture .....	11-12
6.2 UML Diagrams .....	13-24
6.2.1 Class Diagram	
6.2.2 Use Case Diagram	
6.2.3 Sequence Diagram	
6.2.4 Activity Diagram	
6.2.5 Deployment Diagram	
6.2.6 Component Diagram	
6.2.7 ER Diagram	
6.2.8 Data Dictionary Diagram	

## **7. INPUT AND OUTPUT DESIGN**

7.1 Input Design .....	25
7.2 Output Design .....	26

## **8. IMPLEMENTATION**

8.1 Modules .....	27
8.2 Module Description	

## **9. SOFTWARE ENVIRONMENT**

9.1 Java .....	28-55
9.2 Source Code .....	56-64

## **10. RESULT**

10.1 System Testing.....	65
10.2 Output Screens .....	68

## **11. CONCLUSION**

11.1 Conclusion.....	78
11.2 Future Scope	

## **12. REFERENCES .....79**

## LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
6.2.1	CLASS DIAGRAM	15
6.2.2	USE CASE DIAGRAM	14
6.2.3	SEQUENCE DIAGRAM	16
6.2.4	ACTIVITY DIAGRAM	18
6.2.5	DEPLOYMENT DIAGRAM	20
6.2.6	COMPONENT DIAGRAM	19
6.2.7	ER DIAGRAM	21
6.2.8	DATA DICTIONARY	24

## LIST OF SCREENSHOTS

FIGURE NO	FIGURE NAME	PAGE NO
10.1	HOME SCREEN	71
10.2	ADMIN LOGIN	72
10.3	ADMIN LOGIN STATUS	72
10.4	ADMIN HOME	73
10.5	ADD CATEGORY	73
10.6	ADD ALL CATEGORIES	74
10.7	CYBER HARASSERS	74
10.8	USER LOGIN	75
10.9	USER REGISTRATION	75
10.10	USER LOGIN STATUS	76
10.11	USER HOME	76
10.12	POST CONTENT	77
10.13	VIEW ALL POST	77

# **RECOGNITION OF DIGITAL HARASSEMENT ON WEB-BASED SOCIAL CHANNELS**

## **ABSTRACT**

While social media offer great communication opportunities, they also increase the vulnerability of children to threatening situations online. Recent studies report that cyberbullying constitutes a growing problem among children. Successful prevention depends on the adequate detection of potentially harmful messages and the information overload on the Web requires intelligent systems to identify potential risks automatically. 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. Here we have proposed 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



# 1.INTRODUCTION

Web 2.0 has had a substantial impact on communication and relationships in today's society. Children and teenagers go online more frequently, at younger ages, and in more diverse ways (e.g. smartphones, laptops and tablets). Although most of teenagers' Internet use is harmless and the benefits of digital communication are evident, the freedom and anonymity experienced online makes young people vulnerable with cyberbullying being one of the major threats [1, 2].

Bullying is not a new phenomenon and cyberbullying has manifested itself as soon as digital technologies have become primary communication tools. On the positive side, social media like blogs, social networking sites (e.g. Facebook), and instant messaging platforms (e.g. WhatsApp) make it possible to communicate with anyone and at any time. Moreover, they are a place where people engage in social interaction, offering the possibility to establish new relationships and maintain existing friendships [3, 4]. On the negative side however, social media increase the risk of children being confronted with threatening situations including grooming or sexually transgressive behaviour, signals of depression and suicidal thoughts, and cyberbullying. Users are reachable 24/7 and are often able to remain anonymous if desired: this makes social media a convenient way for bullies to target their victims outside the school yard

With regard to cyberbullying, a number of national and international initiatives have been launched over the past few years to increase children's online safety. Examples include *KiVa* (<http://www.kivaprogram.net/>), a Finnish cyberbullying prevention programme, the '*Non au harcèlement*' campaign in France, Belgian governmental initiatives and helplines (e.g. *clicksafe.be*, *veiligonline.be*, *mediawijs.be*) that provide information about online safety, and so on.

In spite of these efforts, a lot of undesirable and hurtful content remains online. [2] analysed a body of quantitative research on cyberbullying and observed cybervictimisation rates among teenagers between 20% and 40%. [5] focused on 12 to 17 year olds living in the United States and found that no less than 72% of them had encountered cyberbullying at least once within the year preceding the questionnaire. [6] surveyed 9 to 26 year olds in the United States, Canada, the United

Kingdom and Australia, and found that 29% of the respondents had ever been victimised online. A study among 2,000 Flemish secondary school students (age 12 to 18) revealed that 11% of them had been bullied online at least once in the six months preceding the survey [7]. Finally, the 2014 large-scale EU Kids Online Report [8] published that 20% of 11 to 16 year olds had been exposed to hate messages online. In addition, youngsters were 12% more likely to be exposed to cyberbullying as compared to 2010, which clearly demonstrates that cyberbullying is a growing problem.

The prevalence of cybervictimisation depends on the conceptualisation used in describing cyberbullying, but also on research variables such as location and the number and age span of the participants. Nevertheless, the above studies demonstrate that online platforms are increasingly used for bullying, which is a cause for concern given its impact. As shown by [9–11], cyberbullying may negatively impact the victim's self-esteem, academic achievement and emotional well-being. [12] found that self-reported effects of cyberbullying include negative effects on school grades and feelings of sadness, anger, fear, and depression. In extreme cases, cyberbullying could even lead to self-harm and suicidal thoughts.

## 2. LITERATURE SURVEY

**Title:** - Child Predator Detection System On Social Media

**Authors:** - Arman Gangani, 2Aman Gangani, 3Shruti Kale, 4Dikshita Jain, 5Prof. A J Kadam

**Abstracts:** - Professional psychologists need to understand the dangers of online sexual harassment and how to protect young people from internet sex predators. While the internet offers many positive aspects, one of the most pernicious issues is its potential use for online sexual exploitation. The internet provides a medium that allows sex predators to access numerous children in a relatively anonymous environment. The main objective of our project is to detect child predators based on comments and posts on social media accounts and send the predator's record to the cyber cell admin. A recent national survey indicated that about one in five youths are solicited for sex over the internet annually (Finkelhor, Mitchell, & Wolak, 2000; Mitchell, Finkelhor, & Wolak, 2001). This project report presents our current development efforts to create this system. As a result, with the developed system, child predator accounts can be detected and reported to the admin for further action.

**Title:** - Child Predator Detection System On Social Media

**Authors:** - anjivani Chavan<sup>1</sup> , Rutuja Konde<sup>2</sup> ,Ishita Rajoria<sup>3</sup> , Tejashree Deshmukh<sup>4\*</sup> , A.S. Sondkar<sup>5</sup>

**Abstracts:** - Increase in Internet use and facilitating access to social media platform has help the predatory to establish online relationships with children which has boost to increase in online solicitation. We are proposing system that enables us to detect a predator in online chats using Text classification method. In this paper, the use of machine learning algorithm named as support vector machine has been used to determine cyber predators. The main objective of our system is to detect child predator base on chat, comments and post of social media account and send predator record to cyber cell admin & the use of PAN12 dataset is done for text classification Purpose. This paper presents our current development to enable the creation of thechild predator system using SVM text classification.

**Title: -** Child Predator Detection in Online Chat Conversation using Support Vector Machine

**Authors: -** Sanjivani Chavan<sup>1</sup> , Rutuja Konde<sup>2</sup> ,Ishita Rajoria<sup>3</sup> , Tejashree Deshmukh<sup>4\*</sup> , A.S. Sondkar<sup>5</sup>

**Abstracts: -** Increase in Internet use and facilitating access to social media platform has help the predatory to establish online relationships with children which has boost to increase in online solicitation. We are proposing system that enables us to detect a predator in online chats using Text classification method. In this paper, the use of machine learning algorithm named as support vector machine has been used to determine cyber predators. The main objective of our system is to detect child predator base on chat, comments and post of social media account and send predator record to cyber cell admin & the use of PAN12 dataset is done for text classification Purpose. This paper presents our current development to enable the creation of thechild predator system using SVM text classification.

## 3.SYSTEM ANALYSIS

### 3.1 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

### DISADVANTAGES

- **Limited Scope of Operation:** Existing systems are mostly limited to gaming environments and audio chats, ignoring other critical platforms like social media, messaging apps, or video-sharing platforms where children spend a large portion of their time.
- **Lack of Text-Based Monitoring:** Most systems focus on voice interactions, neglecting textual conversations, which are often used by predators for grooming and manipulation.
- **No Multimedia Analysis:** These systems usually do not analyze images, videos, or shared media, making them ineffective in detecting visual harassment or explicit content.
- **Platform Dependency:** The detection tools are often platform-specific and do not offer cross-platform protection. Children might be safe in one app but vulnerable in others.

### 3.2 PROPOSED SYSTEM

We propose a child predator detection system that aims to enhance the safety of children in online environments, especially on social media platforms. The system is designed with three core modules: the **User Module**, the **Training Module**, and the **Cyber System**. In the **User Module**, users are categorized into two types—normal users and those exhibiting potential

predator behavior. The **Training Module** employs a machine learning approach using the **Support Vector Machine (SVM) algorithm** for effective text classification and image analysis. This allows the system to intelligently detect suspicious conversations or content shared by users. Once a user is flagged for predator-like behavior, the information is passed on to the **Cyber System**, where all predator reports are reviewed. The cyber administrators can then take appropriate actions such as warnings, account restrictions, or reporting to legal authorities, depending on the severity of the threat.

## **ADVANTAGES**

- **Early Intervention** – By detecting harmful behavior at an early stage, the system helps prevent potential online abuse before it escalates, protecting children from exploitation.
- **Safer Online Spaces** – By continuously monitoring online interactions, the system ensures a more secure and regulated digital environment, minimizing exposure to threats like cyberbullying, grooming, and inappropriate content.
- **Protection of Privacy** – Unlike traditional monitoring tools that may infringe on user privacy, this system focuses specifically on detecting harmful content while maintaining ethical and legal data privacy standards.

## 4. SYSTEM REQUIREMENTS

### 4.1 FUNCTIONAL REQUIREMENTS

A functional requirement is a statement of how a system must behave. It defines what the System should do in order to meet the user's needs or expectations. Functional requirements can be thought of as features that the user detects. They are different from non-functional requirements, which define how the system should work internally (e.g., performance, security, etc.).

These are the modules of

- Admin
- Users

### 4.2 NON – FUNCTIONAL REQUIREMENTS

#### 4.2.1 SOFTWARE REQUIREMENTS:

Operating System	Windows 10/11
Development Software	JAVA 8.1
Programming Language	Java
Integrated Development Environment (IDE)	NetBeans 8.1
Front End Technologies	HTML5, CSS3, Java Script
Database Language	SQL
Database (RDBMS)	MySQL
Database Software	MySQL Server
Web Server or Deployment Server	Apache tomcat
Design/Modelling	Rational Rose

#### 4.2.2 HARDWARE REQUIREMENTS:

MINIMUM (Required for Execution)		MY SYSTEM (Development)
System	Pentium IV 2.2 GHz	I5 Processor 13 <sup>th</sup> Gen
Hard Disk	20 Gb	1 TB
RAM	1 Gb	8 Gb



## **5. SYSTEM STUDY**

### **5.1 FEASIBILITY STUDY:**

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time“. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in there consumption of resources. This place a constant is approval of any project.

- Technical feasibility
- Operational feasibility
- Economical feasibility

### **5.2 FEASIBILITY ANALYSIS**

#### **TECHNICAL FEASIBILITY:**

To determine whether the proposed system is technically feasible, we should take intoconsideration the technical issues involved behind the system. Detection of child predators uses the web technologies, which is rampantly employedthese days worldwide. The world without the web is incomprehensible today. That goes to proposed system is technically feasible.

#### **OPERATIONAL FEASIBILITY:**

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the usersare familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

## **ECONOMIC FEASIBILITY:**

To decide whether a project is economically feasible, we have to consider various factors as Cost benefit analysis Long-term returns Maintenance costs The proposed Detection of child predators is computer based. It requires average computing capabilities and access to internet, which are very basic requirements hence it doesn't incur additional economic overheads, which renders the system economically feasible.

## 6.SYSTEM DESIGN

System design is the transition from a useroriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the systemand format of the fields is also designed. The fields in the database tables should define their rolein the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

## SOFTWARE DESIGN

In designing the software following principles are followed:

1. **Modularity and partitioning:** software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.
2. **Coupling:** modules should have little dependence on other modules of a system.
3. **Cohesion:** modules should carry out in a single processing function.
4. **Shared use:** avoid duplication by allowing a single module is called by other that need the function it provides

## 6.1 SYSTEM ARCHITECTURE

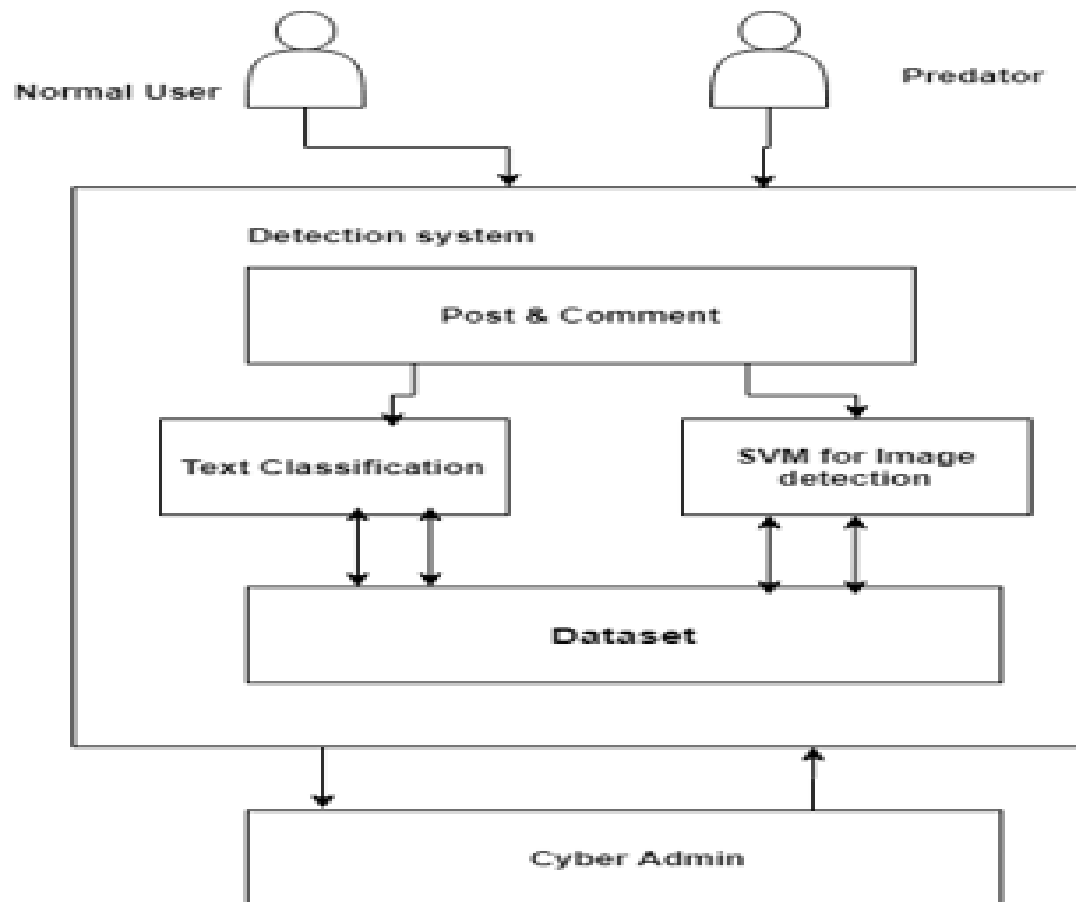


Fig 1. System Architecture

## 6.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

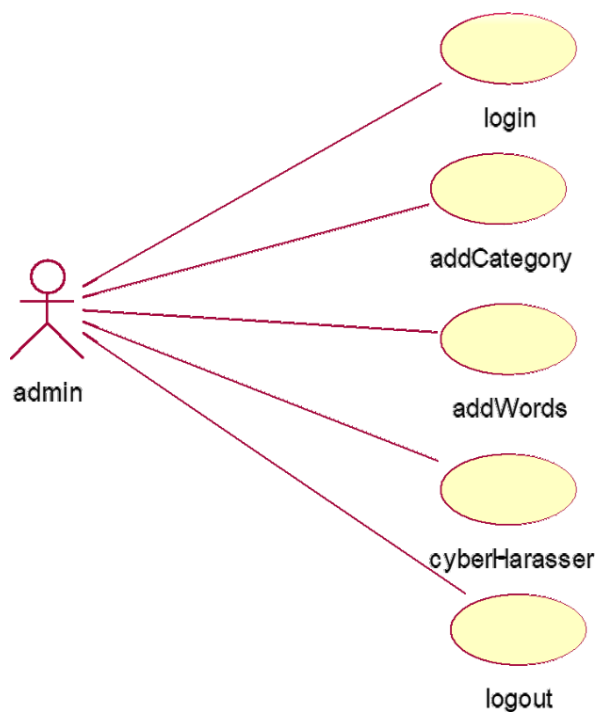
The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

### GOALS:

- The Primary goals in the design of the UML are as follows:
- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process. Provide a formal basis for understanding the modeling language.

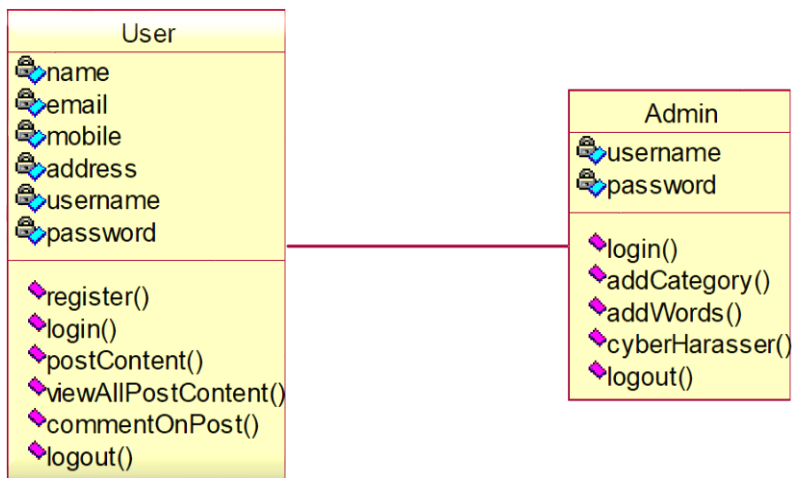
### 6.2.1 USE CASE DIAGRAM:

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. Its 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.



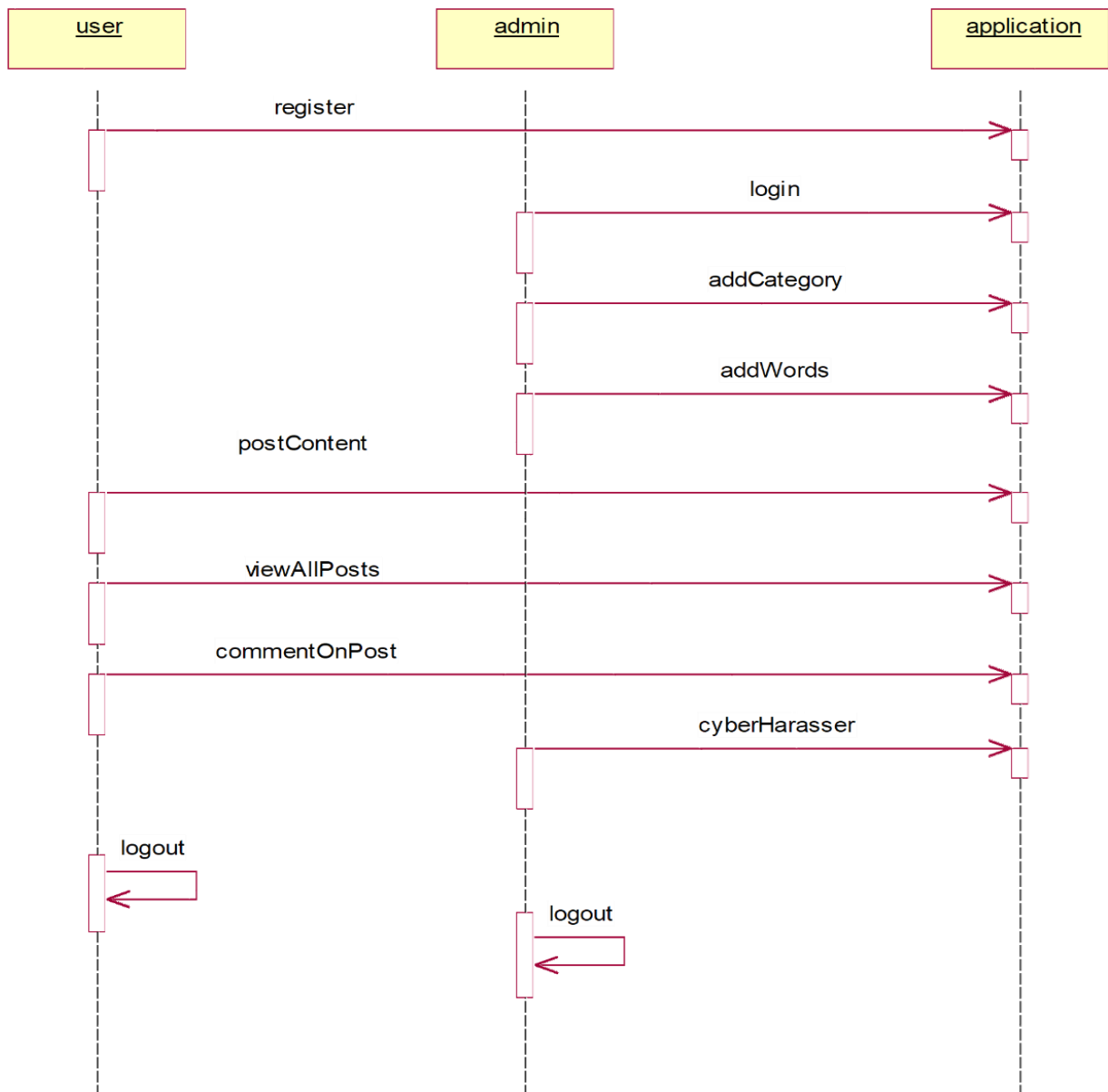
### 6.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



### 6.2.3 SEQUENCE DIAGRAM:

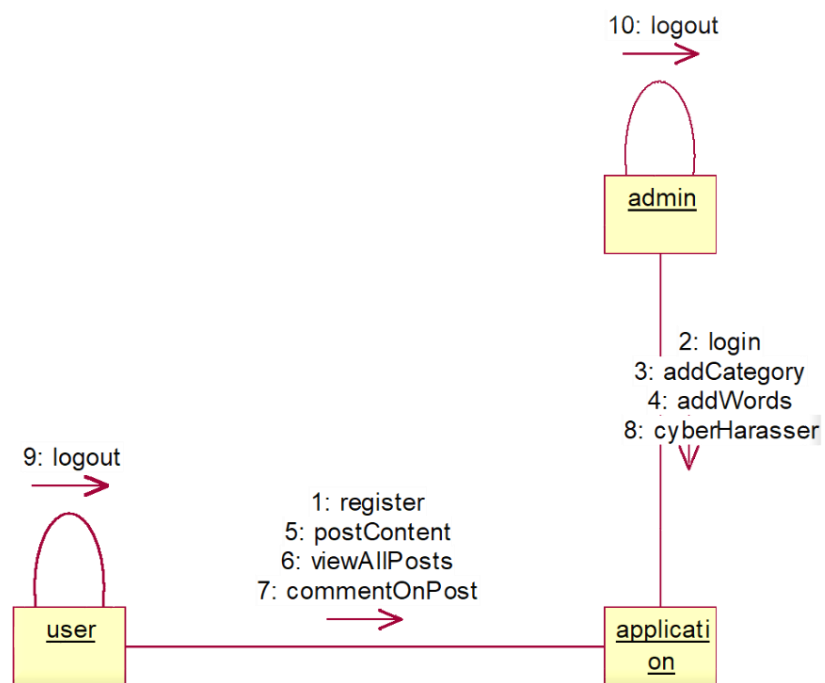
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.





## 6.2.4 COLLABORATION

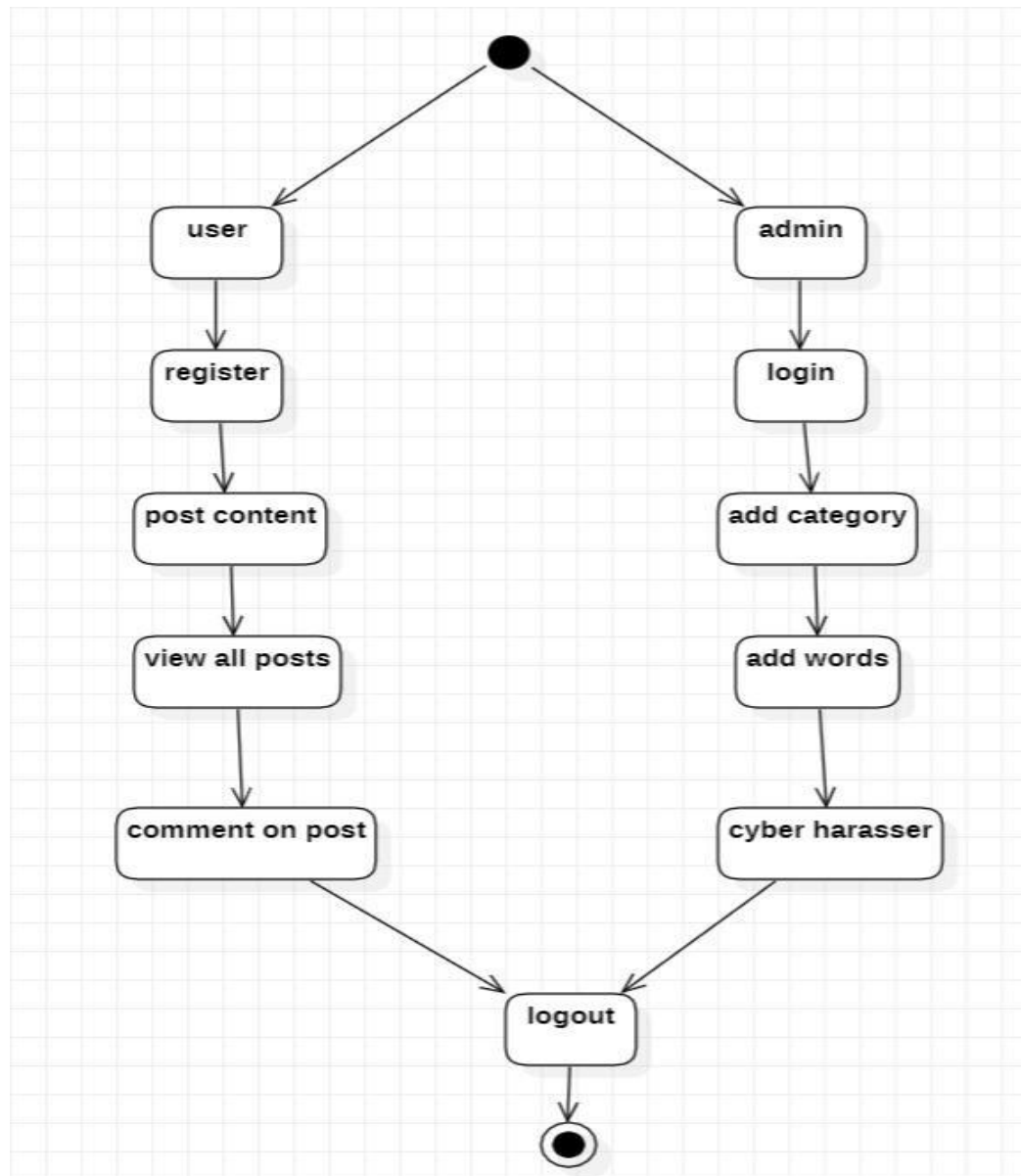
Collaboration is a working practice whereby individuals work together for a common purpose to achieve business benefit. Collaboration enables individuals to work together to achieve a defined and common business purpose.



## 6.2.5 ACTIVITY

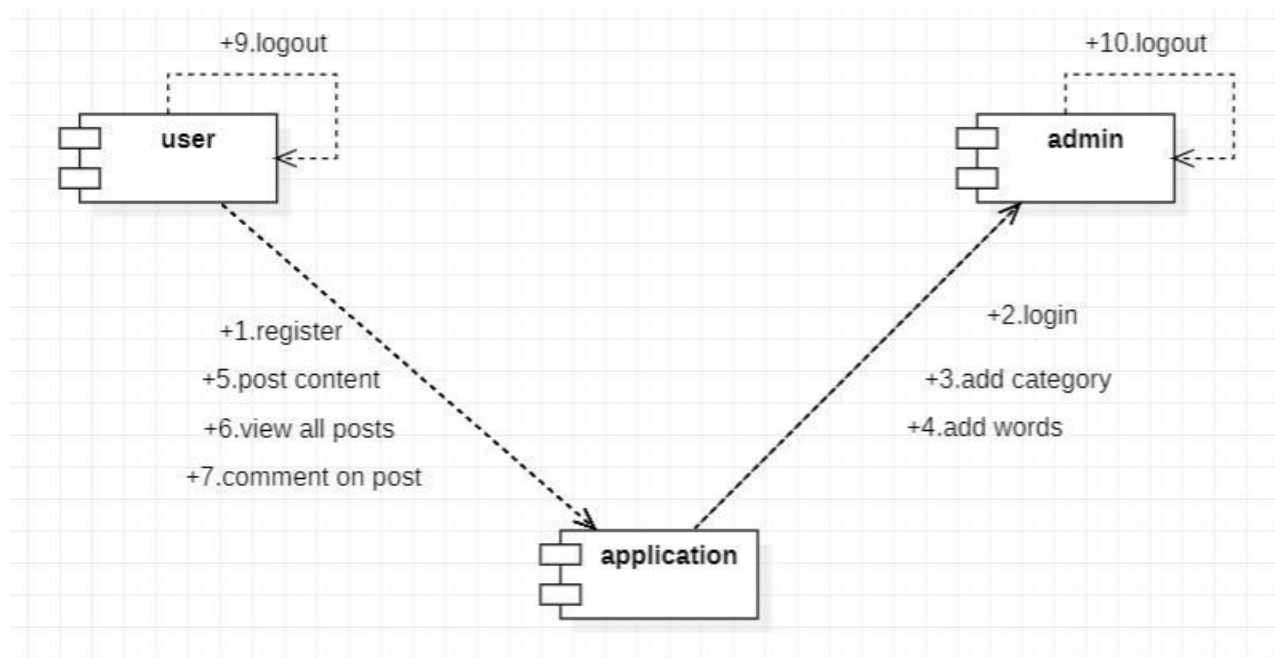
An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram.

Activity diagrams are often used in business process modeling.



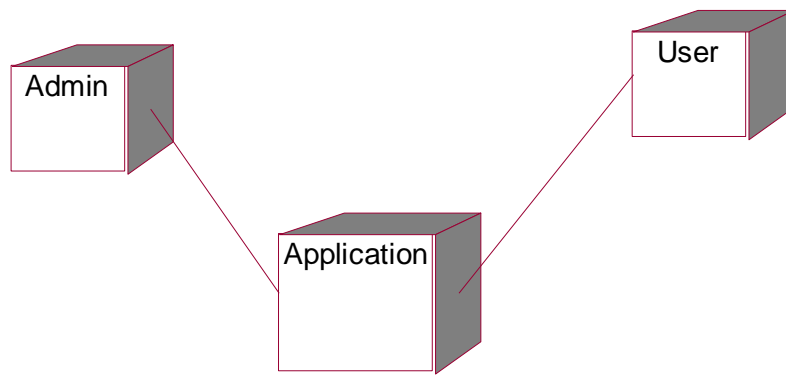
## 6.2.6 COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



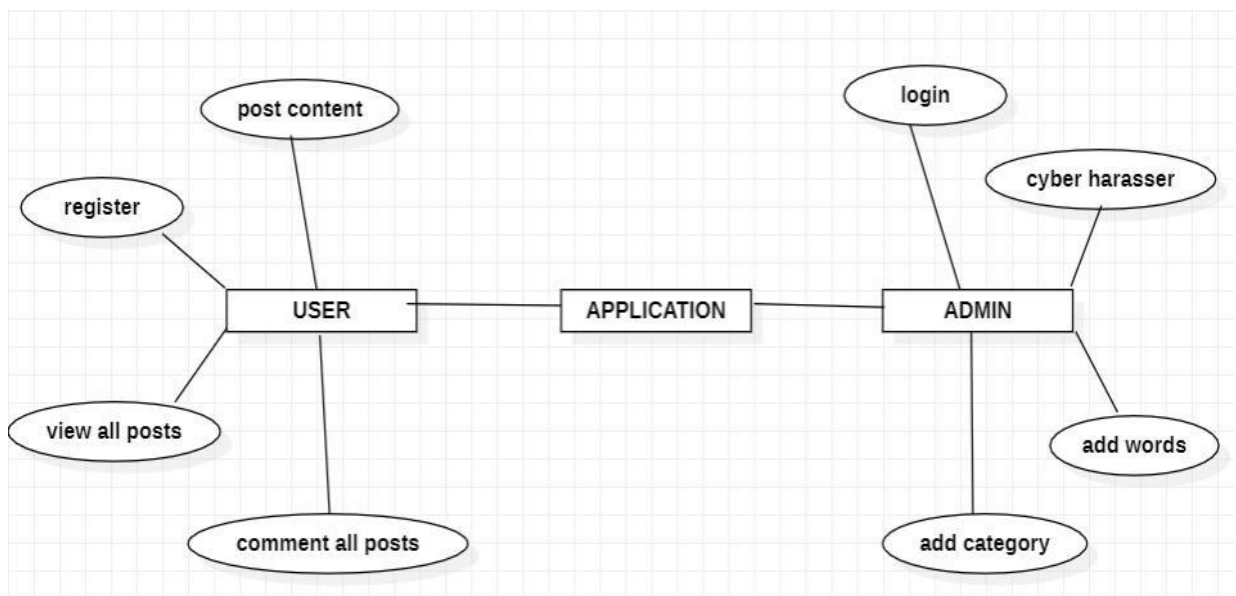
## 6.27 DEPLOYMENT DIAGRAM

A Deployment Diagram in UML (Unified Modeling Language) is used to visualize the physical deployment of software components and hardware nodes within a system or application. It provides a high-level view of how software and hardware interact in a distributed computing environment.



## 6.28 E-R DIAGRAM

An Entity-Relationship Diagram (ERD) is a visual representation used in database design and modeling to illustrate the relationships between entities (also known as tables) in a database. ERDs are a crucial tool for understanding the structure of a database system and how data is organized within it. Here's a description of the key components and concepts associated with an ERD



## 6.2.9 DATA DICTIONARY:

The data dictionary is suitable as a detailed specification of data assets and can be supplemented with ER diagrams, as both serve slightly different purposes.

Core data dictionary elements:

1. List of data sets/tables,
2. List of attributes/columns of each table with data type.

Optional data dictionary elements:

1. Item descriptions,
2. Relationships between tables/columns,
3. Additional constraints, such as uniqueness, default values, value constraints or calculated columns.

## ADMIN:

COLUMN	DATATYPE	CONSTRAINT
username	Varchar(1000)	(NULL)
password	Varchar(1000)	(NULL)

## CATEGORY:

COLUMN	DATATYPE	CONSTRAINT
id	Int(11)	NOTNULL
category	Text	NULL

## COMMENT

COLUMN	DATATYPE	CONSTRAINT
ID	INT(11)	NOTNULL

PID	Text	NULL
UID	Text	NULL
COMMENT	Text	NULL
DATE	Text	NULL

## POSTS

COLUMN	DATATYPE	CONSTRAINT
ID	INT(11)	NOTNULL
NAME	Text	NULL
EMAIL	Text	NULL
MOBILE	Text	NULL
ADDRESS	Text	NULL
USERNAME	Text	NULL
PASSWORD	Text	NULL

## WORDS

COLUMN	DATATYPE	CONSTRAINT
ID	Int(11)	(NOTNULL)

CATEGORY	Varchar(1000)	(NULL)
WORD	Varchar(1000)	(NULL)



## **7.INPUT OUTPUT DESIGNS**

### **7.1 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?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

### **OBJECTIVES**

1. 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.

2. 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.

3. 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

## **7.2 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 intelligent output design improves the system's relationship to help user decision-making.

1. 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.

2. Select methods for presenting information.

3. 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.

Confirm an action.

## **8.IMPLEMENTATION**

### **8.1 MODULES**

- Admin
- User

#### **8.1.1 MODULE DESCRIPTION**

##### **Admin**

In this application the admin is the main module, here admin can directly login with the application no need to register with our application after admin successful login admin can perform some operations such as addcategory, addwords, yberharasser and logout

##### **User**

In this application user is another module here user should register with the application then only user can login with the application. After user successful login he/her can perform some operations such as postContent, viewAllPostContent comment on content and then logout

## **9.SOFTWARE ENVIRONMENT**

### **9.1 JAVA**

#### **HTML**

Html is a language which is used to create web pages with html marking up a page to indicate its format, telling the web browser where you want a new line to begin or how you want text or images aligned and more are possible.

We used the following tags in our project.

#### **TABLE:**

Tables are so popular with web page authors is that they let you arrange the elements of a web page in such a way that the browser won't rearrange them web page authors frequently use tables to structure web pages.

##### **<TR>:**

<TR> is used to create a row in a table encloses <TH> and <TD> elements. <TR> contain many attributes. Some of them are,

ALIGN: specifies the horizontal alignment of the text in the table row.

BGCOLOR: Specifies the background color for the row.

BORDERCOLOR: Sets the external border color for the row.

VALIGN: Sets the vertical alignment of the data in this row.

##### **<TH>:**

<TH> is used to create table heading.

ALIGN: Sets the horizontal alignment of the content in the table cell. Sets LEFT, RIGHT, CENTER.

BACKGROUND: Species the back ground image for the table cell.

BGCOLOR: Specifies the background color of the table cell

## **FORM:**

The purpose of FORM is to create an HTML form; used to enclose HTML controls, like buttons and text fields.

## **ATTRIBUTES:**

**ACTION:** Gives the URL that will handle the form data.

**NAME:** Gives the name to the form so you can reference it in code set to an alphanumeric string.

**METHOD:** method or protocol is used to sending data to the target action URL. The GET method is the default, it is used to send all form name/value pair information in an URL. Using the POST method, the content of the form are encoded as with the GET method, but are sent in environment variables.

## **CONTROLS IN HTML**

### **<INPUT TYPE =BUTTON>:**

Creates an html button in a form.

### **ATTRIBUTES:**

**NAME:** gives the element a name. Set to alphanumeric characters.

**SIZE:** sets the size.

**VALUE:** sets the caption of the element.

### **<INPUT TYPE = PASSWORD>:**

Creates a password text field, which makes typed input.

### **ATTRIBUTES:**

**NAME:** gives the element a name, set to alphanumeric characters.

**VALUE:** sets the default content of the element.

### **<INPUT TYPE=RADIO>:**

Creates a radio button in a form.

**ATTRIBUTE:**

**NAME:** Gives the element a name. Set to alphanumeric character.

**VALUE:** Sets the default content of the element.

**<INPUT TYPE=SUBMIT>:**

Creates a submit button that the user can click to send data in the form back to the web server.

**ATTRIBUTES:**

**NAME:** Gives the element a name. Set to alphanumeric characters.

**VALUE:** Gives this button another label besides the default, Submit Query. Set to alphanumeric characters.

**<INPUT TYPE=TEXT>:**

Creates a text field that the user can enter or edit text in.

**ATTRIBUTES:**

**NAME:** Gives the element a name. Set to alphanumeric characters.

**VALUE:** Holds the initial text in the text field. Set to alphanumeric characters

## **JAVA SCRIPT**

Java script originally supported by Netscape navigator is the most popular web scripting language today. Java script lets you embedded programs right in your web pages and run these programs using the web browser. You place these programs in a <SCRIPT> element, usually with in the <HEAD> element. If you want the script to write directly to the webpage, place it in the <BODY> element.

## **JAVASCRIPT METHODS:**

### **Writeln:**

Document.writeln() is a method, which is used to write some text to the current web page.

### **onClick:**

Occurs when an element is clicked.

### **onLoad:**

Occurs when the page loads.

### **onMouseDown:**

Occurs when a mouse button goes down.

### **onMouseMove:**

Occurs when the mouse moves.

### **onUnload:**

Occurs when a page is unloaded.

## MySQL

The database has become an integral part of almost every human's life. Without it, many things we do would become very tedious, perhaps impossible tasks. Banks, universities, and libraries are three examples of organizations that depend heavily on some sort of database system. On the Internet, search engines, online gas booking , and even the website naming convention (<http://www...>) would be impossible without the use of a database. A database that is implemented and interfaced on a computer is often termed a database server.

One of the fastest SQL (Structured Query Language) database servers currently on the market is the MySQL server. MySQL, available for download, offers the database programmer with an array of options and capabilities rarely seen in other database servers. What's more, MySQL is free of charge for those wishing to use it for private and commercial use. Those wishing to develop applications specifically using MySQL should consult MySQL's licensing section, as there is a charge for licensing the product.

These capabilities range across a number of topics, including the following:

Ability to handle an unlimited number of simultaneous users.

Capacity to handle 50,000,000+ records.

Very fast command execution, perhaps the fastest to be found on the market.

Easy and efficient user privilege system.

A database is really nothing more than a hierarchy of increasingly complex data structures. In MySQL, the acknowledged structure for holding blocks (or **records**) of information is called the **table**.

These records, in turn, are made up of the smallest object that can be manipulated by the user, known as the **data type**. Together, one or more of these data types form a record. A table holds the collection of records that make up part of the database. We can consider the hierarchy of a database to be that of the following:

Database < Table < Record < Datatype



Datatypes come in several forms and sizes, allowing the programmer to create tables suited for the scope of the project. The decisions made in choosing proper data types greatly influence the performance of a database, so it is wise to have a detailed understanding of these concepts.

## **MySQL Data types**

MySQL is capable of many of the data types that even the novice programmer has probably already been exposed to. Some of the more commonly used include:

### **CHAR (M)**

CHAR's are used to represent fixed length strings. A CHAR string can range from 1-255 characters. In later table creation, an example CHAR data type would be declared as follows:

ex.

```
car model CHAR(10);
```

### **VARCHAR (M)**

VARCHAR is a more flexible form of the CHAR data type. It also represents data of type String, yet stores this data in variable length format. Again, VARCHAR can hold 1-255 characters.

VARCHAR is usually a wiser choice than CHAR, due to it's variable length format characteristic. Although, keep in mind that CHAR is much faster than VARCHAR, sometimes up to 50%.

(A CHAR stores the whole length of the declared variable, regardless of the size of the data contained within, whereas a VARCHAR only stores the length of the data, thus reducing size of the database file.) ex.

```
car model VARCHAR(10);
```

### **INT (M) [Unsigned]**

The INT data type stores integers ranging from -2147483648 to 2147483647. An optional "unsigned" can be denoted with the declaration, modifying the range to be 0 to 4294967295

ex.

```
light-years INT;
```

Valid integer: '-24567'. Invalid integer: '30000000000'.

ex.

```
light-years INT unsigned;
```

Valid integer: '30000000000'. Invalid integer: '-24567'.

## **FLOAT [(M,D)]**

A FLOAT represents small decimal numbers, used when a somewhat more precise representation of a number is required.

ex.

rainfall FLOAT (4,2);

This could be used to represent rainfall average in centimeters per year, which could be a decimal value. More specifically, FLOAT (4,2) states the fact that rainfall can hold up to four characters and two decimal places. Thus,

42.35 is valid, accurately represented.

324.45 is invalid, rounded to 324.5.

2.2 is valid, accurately represented.

34.542 is invalid, rounded to 34.54.

*Note: Due to the fact that FLOAT is rounded, those wishing to represent money values would find it wise to use **DECIMAL**, a datatype found within MySQL that does not round values. Consult the documentation for a complete explanation.*

## **DATE**

Stores date related information. The default format is 'YYYY-MM-DD', and ranges from '0000-00-00' to '9999-12-31'. MySQL provides a powerful set of date formatting and manipulation commands, too numerous to be covered within this article. However, one can find these functions covered in detail within the MySQL documentation.

the\_date DATE;

## **TEXT / BLOB**

The text and blob datatypes are used when a string of 255 - 65535 characters is required to be stored. This is useful when one would need to store an article such as the one you are reading. However, there is no end space truncation as with VARCHAR AND CHAR. The only difference between BLOB and TEXT is that TEXT is compared case insensitively, while BLOB is compared case sensitively.

## **SET**

A datatype of type string that allows one to choose from a designated set of values, be it one value or several values. One can designate up to 64 values.

ex.

```
transport SET ("truck", "wagon") NOT NULL;
```

From the above declaration, the following values can be held by transport:

```
""  
"truck"  
"wagon"  
"truck,wagon"
```

## **ENUM**

A datatype of type string that has the same characteristics as the SET datatype, but only one set of allowed values may be chosen. Usually only takes up one byte of space, thus saving time and space within a table.

ex.

```
transport ENUM ("truck", "wagon") NOT NULL;
```

From the above declaration, the following values can be held by transport:

```
""  
"truck"  
"wagon"
```

## **Records**

Together, a group of declared datatypes form what is known as a record. A record can be as small as one data variable, or as many as deemed needed. One or more records form the structure of a table.

## The Bigger Picture: Tables

Before we can execute commands on the database, we must first create a table in which data can be stored. This is accomplished in the following manner:

```
mysql> CREATE TABLE test (  
> name VARCHAR (15),  
> email VARCHAR (25),  
> phone_number INT,  
> ID INT NOT NULL AUTO_INCREMENT,  
> PRIMARY KEY (ID));
```

Ensuing output:

```
Query OK, 0 rows affected (0.10 sec)  
mysql>
```

The first table in your database has now been created. *Note: no two tables can have the same name.*

*Note(2): Each dataspace is more often referred to as a **column**.*

### Column Characteristics:

A name may not be made up of strictly numbers.

A name may start with a number.

A name may be up to 64 characters.

### Other table options:

The following options can be placed after any datatype, adding other characteristics and capabilities to them.

Primary Key. Used to differentiate one record from another. No two records can have the same primary key. This is obviously useful when it is imperative that no two records are mistaken to be the other.

Auto\_Increment. A column with this function is automatically incremented one value (previous + 1) when an insertion is made into the record. The datatype is automatically incremented when 'NULL' is inserted into the column.

NOT NULL. Signifies that the column can never be assigned a NULL value.

ex.

```
soc_sec_number INT PRIMARY KEY;
```

No two soc\_sec\_number records can hold the same value.

```
ID_NUMBER INT AUTO_INCREMENT;
```

### Insertion of records

**Note:** The originally created table, test, created in the last section will be used to illustrate the examples in this section. Here it is again, for quick reference:

---

```
mysql> CREATE TABLE test (  
> name VARCHAR (15),  
> email VARCHAR (25),  
> phone_number INT,  
> ID INT NOT NULL AUTO_INCREMENT,  
> PRIMARY KEY (ID));
```

---

Insertion of data into the table is accomplished, logically enough, using the INSERT command.

---

```
mysql> INSERT INTO test VALUES
```

```
mysql> ('Bugs Bunny', 'carrots@devshed.com',
```

```
mysql> 5554321, NULL);
```

---

Result, assuming the command was correctly entered:

---

Query OK, 1 row affected (0.02 sec)

```
mysql>
```

---

## Selection

A database would not be much use if one was not able to search and extract data from it. In MySQL terms, this is accomplished through the SELECT statement.

---

```
mysql> SELECT * FROM test
```

```
mysql> WHERE (name = "Bugs Bunny");
```

---

Result:

name	email	phone	ID
Bugs Bunny	carrots@devshed.com	5554321	1

Let's assume we have inserted four differing records, all bearing the same name of "Bugs Bunny", yet having different email addresses and phone numbers. The table test, would look somewhat like the following:

name	email	phone	ID
------	-------	-------	----

Bugs Bunny	carrots@devshed.com	5554321	1
Bugs Bunny	peppers@devshed.com	5554331	2
Bugs Bunny	lettuce@devshed.com	5554341	3
Bugs Bunny	celery@devshed.com	5554351	4

## Deletion

One can also delete records inserted into the table. This is accomplished through the DELETE command.

---

```
mysql> DELETE FROM test
```

```
mysql> WHERE (name = "Bugs Bunny");
```

---

Result:

This would result in the deletion of all records within the table test containing name "Bugs Bunny".

Another example:

---

```
mysql> DELETE FROM test
```

```
mysql> WHERE (phone_number = 5554321);
```

---

Result: (Using the previously illustrated example)

name	email	phone	ID
Bugs Bunny	peppers@devshed.com	5554331	2
Bugs Bunny	lettuce@devshed.com	5554341	3
Bugs Bunny	celery@devshed.com	5554351	4

## Modification

MySQL also has the capability of modifying data already entered into the table. This is accomplished through the UPDATE command.

---

```
mysql> UPDATE test SET name = 'Daffy Duck'
```

```
mysql> WHERE name = "Bugs Bunny";
```

---

name	email	phone	ID
Daffy Duck	peppers@devshed.com	5554331	2
Daffy Duck	lettuce@devshed.com	5554341	3
Daffy Duck	celery@devshed.com	5554351	4

This section, we covered the core [MySQL database](#) manipulation functions, basic insertion, deletion, modification, and search. The next section will elaborate on these capabilities, providing extended functioning and flexibility when manipulating the database.

What we have covered so far is but a small part of what MySQL is capable of. Let's delve a little deeper into the language, exploring some of the more advanced commands of the language.



## Logical Operations

MySQL includes full support of all basic logical operations.

### AND (&&)

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name = "Bugs Bunny") AND  
mysql> (phone_number = 5554321);
```

---

Result:

All records containing the name "Bugs Bunny" AND the phone number '5554321' will be displayed to the screen.

### OR ( || )

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name = "Bugs Bunny") OR  
mysql> (phone_number = 5554321);
```

---

Result:

All records containing the name "Bugs Bunny" OR the phone number '5554321' will be displayed to the screen.

### NOT ( ! )

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name != "Bugs Bunny");
```

---

Result:

All records NOT containing the name "Bugs Bunny" will be displayed to the screen.

## Order By

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name = "Bugs Bunny") ORDER BY  
mysql> phone_number;
```

---

### Result:

All records containing the name "Bugs Bunny" will be displayed to the screen, ordered in respect to the phone\_number.

## Search functions

MySQL offers the user the ability to perform both general and specific searches on data.

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name LIKE "%gs Bunny");
```

---

### Result:

All records containing the partial string "gs Bunny" will be displayed to the screen. This would include such names as: "Bugs Bunny", "ags Bunny", "gs Bunny", and "234rtgs Bunny".

Notice that "LIKE" has been used instead of the equals sign (=). "LIKE" signifies that one is searching for an estimate of the data requested, and not necessarily an exact copy.

The '%' sign could be placed anywhere within the string. The method in which the server searches for a string is dependent upon where one places the '%' sign.

---

```
mysql> SELECT * FROM test WHERE  
mysql> (name LIKE "Bugs Bunny%");
```

---

### Result:

All records containing the partial string "Bugs Bunny" will be displayed to the screen. This would include such names as: "Bugs Bunnys", "Bugs Bunnyyyy453", "Bugs Bunnytrtrtrtrtr", but not "gs Bunny".

## Focused Search Results

One can also perform searches and display only certain columns.

---

```
mysql> SELECT name FROM test WHERE
```

```
mysql> (name = "Bugs Bunny");
```

---

Result:

name
Bugs Bunny

## Alter table

Another very important function of MySQL is the ability to modify previously created tables. This is accomplished via the ALTER statement. This function allows one to add, modify, and delete columns, as well as rename the table, among other functions.

Example: Rename the table

---

```
mysql> ALTER table test RENAME mytest;
```

---

Example: Add a column

---

```
mysql> ALTER table mytest ADD birthday DATE;
```

---

Example: Modify a column

---

```
mysql> ALTER table mytest CHANGE
```

```
mysql> name newname VARCHAR (25);
```

---

Example: Delete a column

---

```
mysql> ALTER table mytest DROP newname;
```

---

Executing the above four functions would modify test, creating the following table:

---

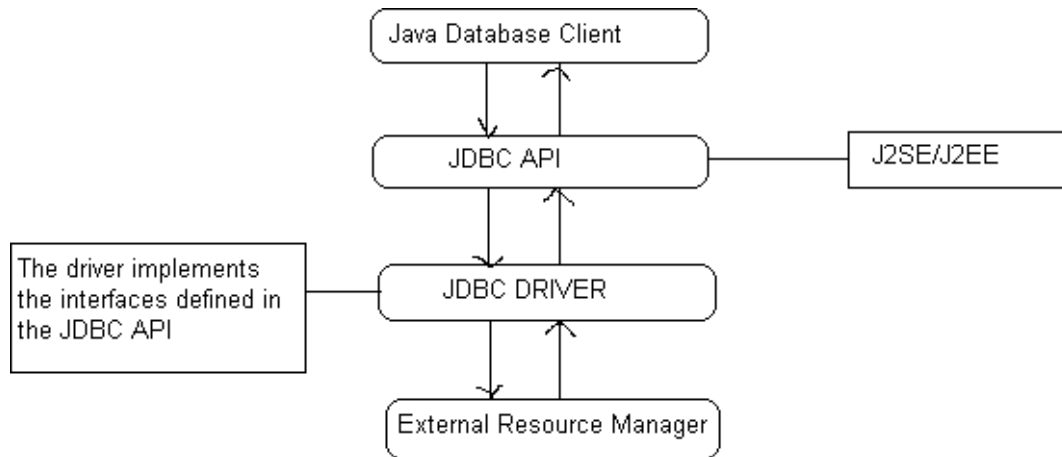
```
mysql> TABLE mytest (
```

```
> email VARCHAR (25),
```

```
> phone_number INT,
```

```
> ID INT AUTO_INCREMENT,
```

```
> birthday DATE );
```



### **DRIVER MANAGER AND DRIVER:**

The `java.sql` package defines an interface called `Java.sql.Driver` that makes to be implemented by all the JDBC drivers and a class called `java.sql.DriverManager` that acts as the interface to the database clients for performing tasks like connecting to external resource managers, and setting log streams. When a JDBC client requests the `DriverManager` to make a connection to an external resource manager, it delegates the task to an appropriate driver class implemented by the JDBC driver provided either by the resource manager vendor or a thirdparty.

### **JAVA.SQL.DRIVERMANAGER:**

The primary task of the class driver manager is to manage the various JDBC drivers register. It also provides methods for:

Getting connections to the databases.

Managing JDBC logs.

Setting login timeout.

## MANAGING DRIVERS:

JDBC clients specify the JDBC URL when they request a connection. The driver manager can find a driver that matches the request URL from the list of register drivers and delegate the connection request to that driver if it finds a match JDBC URLs normally take the following format:

**<protocol>:<sub-protocol>:<resource>**

The protocol is always jdbc and the sub-protocol and resource depend on the type of resource manager. The URL for postgresSQL is in the format:

**Jdbc: postgres ://< host> :< port>/<database>**

Here host is the host address on which post master is running and database is the name of the database to which the client wishes to connect.

## MANAGING CONNECTION:

DriverManager class is responsible for managing connections to the databases:

public static Connection getConnection (String url, Properties info) throws SQLException

This method gets a connection to the database by the specified JDBC URL using the specified username and password. This method throws an instance of SQLException if a database access error occurs.

## CONNECTIONS:

The interface java.sql.Connection defines the methods required for a persistent connection to the database. The JDBC driver vendor implements this interface. A database 'vendor-neutral' client never uses the implementation class and will always use only the interface.

This interface defines methods for the following tasks:

Statements, prepared statements, and callable statements are the different types of statements for issuing sql statements to the database by the JDBC clients.

For getting and setting auto-commit mode.

Getting meta information about the database.

Committing and rolling back transactions.

### **CREATING STATEMENTS:**

The interface `java.sql.Connection` defines a set of methods for creating database statements. Database statements are used for sending SQL statements to the database:

`Public Statement createStatement () throws SQLException`

This method is used for creating instances of the interface `java.sql.Statement`. This interface can be used for sending SQL statements to the database. The interface `java.sql.Statement` is normally used for sending SQL statements that don't take any arguments. This method throws an instance of `SQLException` if a database access error occur:

`Public Statement createStatement (int resType, int resConcurrency) throws SQLException`

### **JDBC RESULTSETS:**

A JDBC resultset represents a two dimensional array of data produced as a result of executing SQL SELECT statements against databases using JDBC statements. JDBC resultsets are represented by the interface `java.sql.ResultSet`. The JDBC vendor provider provides the implementation class for this interface.

### **SCROLLING RESULTSETS:**

`public boolean next() throws SQLException`

`public boolean previous() throws SQLException`

`public boolean first() throws SQLException`

`public boolean last() throws SQLException`

## STATEMENT:

The interface `java.sql.Statement` is normally used for sending SQL statements that do not have IN or OUT parameters. The JDBC driver vendor provides the implementation class for this interface. The common methods required by the different JDBC statements are defined in this interface. The methods defined by `java.sql.Statement` can be broadly categorized as follows:

Executing SQL statements

Querying results and resultsets

Handling SQL batches

Other miscellaneous methods

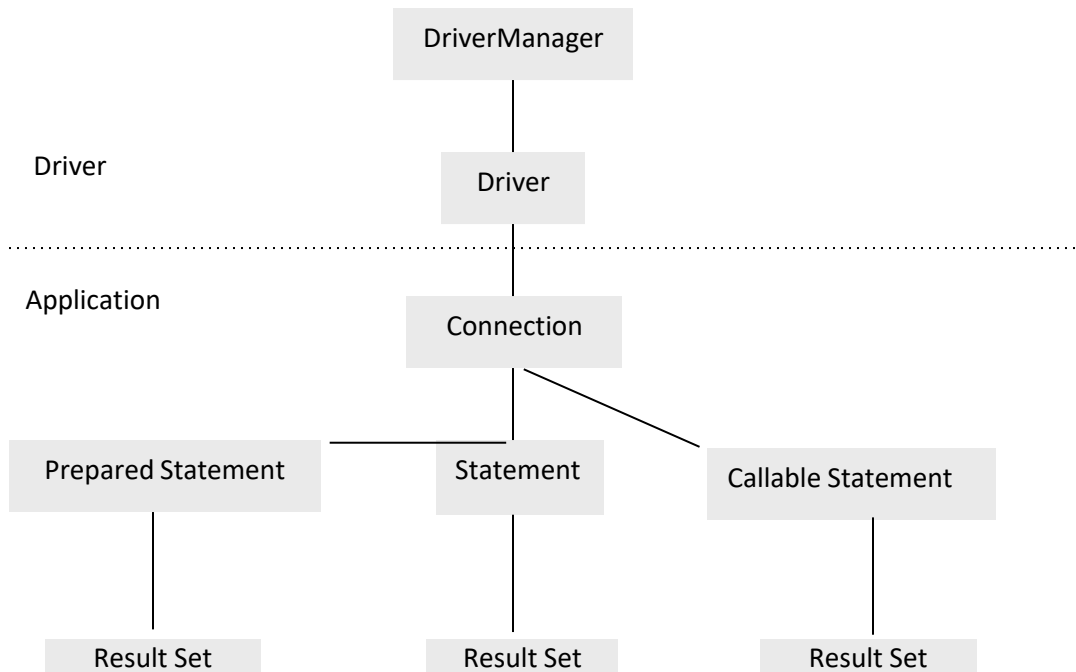
The interface `java.sql.Statement` defines

methods for executing different SQL statements like `SELECT`, `UPDATE`, `INSERT`, `DELETE`, and `CREATE`.

`public ResultSet executeQuery (String sql) throws SQLException`

The following figure shows how the `DriverManager`, `Driver`, `Connection`, `Statement`, `ResultSet` classes are connected.





## **JAVA SERVER PAGES (JSP)**

### **INTRODUCTION:**

Java Server Pages (JSP) technology enables you to mix regular, static HTML with dynamically generated content. You simply write the regular HTML in the normal manner, using familiar Web-page-building tools. You then enclose the code for the dynamic parts in special tags, most of which start with `<%` and end with `%>`.

### **THE NEED FOR JSP:**

Servlets are indeed useful, and JSP by no means makes them obsolete. However,

It is hard to write and maintain the HTML.

You cannot use standard HTML tools.

The HTML is inaccessible to non-Java developers.

## **BENEFITS OF JSP:**

JSP provides the following benefits over servlets alone:

- It is easier to write and maintain the HTML: In this no extra backslashes, no double quotes, and no lurking Java syntax.

You can use standard Web-site development tools:

- We use Macromedia Dreamweaver for most of the JSP pages. Even HTML tools that know nothing about JSP can be used because they simply ignore the JSP tags.

You can divide up your development team:

- The Java programmers can work on the dynamic code. The Web developers can concatenate on the representation layer. On large projects, this division is very important.
- Depending on the size of your team and the complexity of your project, you can enforce a weaker or stronger separation between the static HTML and the dynamic content.

## **CREATING TEMPLATE TEXT:**

A large percentage of our JSP document consists of static text known as template text. In almost all respects, this HTML looks just like normal HTML follows all the same syntax rules, and simply “passed through” to that client by the servlet created to handle the page. Not only does the HTML look normal, it can be created by whatever tools you already are using for building Web pages.

There are two minor exceptions to the “template text passed through” rule.

First, if you want to have `<% Or %>` in the output, you need to put `<\% or %\>` in the template text. Second, if you want a comment to appear in the JSP page but not in the resultant document,

`<%-- JSP Comment -- %>`

HTML comments of the form:

`<!--HTML Comment -->`

are passed through to the client normally.

## **TYPES OF JSP SCRIPTING ELEMENTS:**

JSP scripting elements allow you to insert Java code into the servlet that will be generated from the JSP page. There are three forms:

**Expressions** of the form `<%=Java Expression %>`, which are evaluated and inserted into the servlet's output.

**Scriptlets** of the form `<%Java code %>`, which are inserted into the servlet's `_jspService` method (called by service).

**Declarations** of the form `<%! Field/Method Declaration %>`, which are inserted into the body of the servlet class, outside any existing methods.

## **USING JSP EXPRESSIONS:**

A JSP element is used to insert values directly into the output. It has the following form:

`<%= Java Expression %>`

The expression is evaluated, converted to a string, and inserted in the page. This evaluation is performed at runtime (when the page is requested) and thus has full access to the information about the request. For example, the following shows the date/time that the page was requested.

Current time: `<%=new java.util.Date () %>`

## **PREDEFINED VARIABLES:**

To simplify expressions we can use a number of predefined variables (or “implicit objects”). The specialty of these variables is that, the system simply tells what names it will use for the local variables in `_jspService`. The most important ones of these are:

**request**, the `HttpServletRequest`.

**response**, the `HttpServletResponse`.

**session**, the `HttpSession` associated with the request

**out**, the writer used to send output to clients.

**application**, the `ServletContext`. This is a data structure shared by all servlets and JSP pages in the web application and is good for storing shared data.

Here is an example:

Your hostname: `<%= request.getRemoteHost () %>`

## COMPARING SERVLETS TO JSP PAGES

JSP works best when the structure of the HTML page is fixed but the values at various places need to be computed dynamically. If the structure of the page is dynamic, JSP is less beneficial. Some times servlets are better in such a case. If the page consists of binary data or has little static content, servlets are clearly superior. Sometimes the answer is neither servlets nor JSP alone, but rather a combination of both.

## WRITING SCRIPTLETS

If you want to do something more complex than output the value of a simple expression .JSP scriptlets let you insert arbitrary code into the servlet's `_jspService` method. Scriptlets have the following form:

`<% Java code %>`

Scriptlets have access to the same automatically defined variables as do expressions (`request`, `response`, `session`, `out` , etc ) .So for example you want to explicitly send output of the resultant page , you could use the `out` variable , as in the following example:

```

<%

String queryData = request.getQueryString ();

out.println ("Attached GET data: "+ queryData);

%>

```

## JAKARTA TOMCAT

2-713 Tomcat is the Servlet/JSP container. Tomcat implements the Servlet 2.4 and Java Server Pages 2.0 specification. It also includes many additional features that make it a useful platform for developing and deploying web applications and web services.

### TERMINOLOGY:

Context – a Context is a web application.

\$CATALINA\_HOME – This represents the root of Tomcat installation.

### DIRECTORIES AND FILES:

**/bin** – Startup, shutdown, and other scripts. The \*.sh files (for Unix systems) are functional duplicates of the \*.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.

**/conf** – Configuration files and related DTDs. The most important file in here is server.xml. It is the main configuration file for the container.

**/logs** – Log files are here by default.

**/webapps** – This is where webapps go\

## **INSTALLATION:**

Tomcat will operate under any Java Development Kit (JDK) environment that provides a JDK 1.2 (also known as Java2 Standard Edition, or J2SE) or later platform. JDK is needed so that servlets, other classes, and JSP pages can be compiled.

## **DEPLOYMENT DIRECTORIES FOR DEFAULT WEB APPLICATION:**

### **HTML and JSP Files**

Main Location

\$CATALINA\_HOME/webapps/ROOT

Corresponding URLs.

<http://host/SomeFile.html>

<http://host/SomeFile.jsp>

More Specific Location (Arbitrary Subdirectory).

\$CATALINA\_HOME/webapps/ROOT/SomeDirectory

Corresponding URLs

<http://host/SomeDirectory/SomeFile.html>

<http://host/SomeDirectory/SomeFile.jsp>

### **Individual Servlet and Utility Class Files**

Main Location (Classes without Packages).

\$CATALINA\_HOME/webapps/ROOT/WEB-INF/classes

Corresponding URL (Servlets).

<http://host/servlet/ServletName>

More Specific Location (Classes in Packages).

\$CATALINA\_HOME/webapps/ROOT/WEB-INF/classes/packageName

Corresponding URL (Servlets in Packages).

http://host/servlet/packageName.ServletName

## **Servlet and Utility Class Files Bundled in JAR Files**

Location

\$CATALINA\_HOME/webapps/ROOT/WEB-INF/lib

Corresponding URLs (Servlets)

http://host/servlet/ServletName

## **9.2 SOURCE CODE**

### **ADMIN.JSP**

<%--

Document : Admin

Created on : 7 Feb, 2021, 4:04:29 PM

Author : KishanVenky

--%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Detections</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />



```

</head>

<body id="top">

<div class="wrapper col1">

  <div id="header">

    <div id="logo">

      <h1><a href="#">Detection of Cyber Harassers </a></h1>

      <p>on Social Media</p>

    </div>

    <div id="info">


    </div>

    <br class="clear" />

  </div>

</div>

<!--
#####
##### -->

<div class="wrapper col2">

  <div id="topbar">

    <div id="topnav">

      <ul>

        <li><a href="index.html">Home</a></li>

```

```

<li class="active"><a href="Admin.jsp">Admin</a></li>

<li><a href="User.jsp">User</a></li>


<li class="last"><a href="Register.jsp">Registration</a></li>

</ul>

</div>

<div id="search">


</div>

<br class="clear" />

</div>

</div>

<!--
#####
##### -->

<div class="wrapper col3">

<div id="intro">

<div class="fl_left"><a href="#"></a></div>

<div class="fl_right">

<h2>About Project</h2>
<p>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. • Desired Solution : 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</p>

</div>

<br class="clear" />

</div>

</div>

<!--

#####

##### -->

<div class="wrapper col4">

<div id="container">

<div id="services">

<br class="clear" />

</div>

<div id="content">

<center>

<h2>ADMIN LOGIN</h2>

<form action="AdminLAction.jsp" method="post">

<table>

```
<tr><th>UserName</th><td><input type="text" name="uname"
required=""></td></tr>
```

```
<tr><th>Password</th><td><input type="password" name="pass"
required=""></td></tr>
```

```
<tr><th></th><td><input type="submit" value="Login">
```

```
</td></tr>
```

```
</table>
```

```
</form>
```

```
</center>
```

```
</div>
```

```
<div id="column">
```

```
<div class="flickrbox">
```

```
<br class="clear" />
```

```
</div>
```

```
</div>
```

```
<br class="clear" />
```

```
</div>
```

```
</div>
```

```
<!--
```

```
#####
```

```
##### -->
```

```
<div class="wrapper col5">
```

```

<div id="footer">

    <br class="clear" />

</div>

</div>

<!--
#####
##### -->

<div class="wrapper col6">

    <div id="copyright">

        <p class="fl_left">Detection of Cyber Harassers on Social Media</p>

        <p class="fl_right"></p>

        <br class="clear" />

    </div>

</div>

</body>

</html>

```

## **USER.JSP**

```
@charset "utf-8";
```

```
/*
```

Template Name: BusinessBlue

Author: <a href="http://www.os-templates.com/">OS Templates</a>

Author URI: <http://www.os-templates.com/>

Licence: Free to use under our free template licence terms

Licence URI: <http://www.os-templates.com/template-terms>

File: Layout CSS

\*/

```
@import url("navi.css");
```

```
@import url("forms.css");
```

```
@import url("tables.css");
```

```
html{overflow-y:scroll;}
```

```
body{margin:0; padding:0; font-size:12px; font-family:verdana, Arial, Helvetica, sans-serif;
color:#FFFFFF; background-color:#1A3151;}
```

```
.justify{ text-align:justify;}
```

```
.bold{ font-weight:bold;}
```

```
.center{ text-align:center;}
```

```
.right{ text-align:right;}
```

```
.nostart{ margin:0; padding:0; list-style-type:none;}
```

```
.clear{ clear:both;}
```

```
br.clear{ clear:both; margin-top:-15px;}
```

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

```
.fl_left, .imgl{float:left;}
```

```
.fl_right, .imgr{float:right;}
```

```
img{display:block; margin:0; padding:0; border:none;}
```

```
.imgl, .imgr{border:1px solid #CCCCCC; padding:5px;}
```

```
.imgl{margin:0 8px 8px 0; clear:left;}
```

```
.imgr{margin:0 0 8px 8px; clear:right;}
```

```
/* .....Header..... */
```

```
#header{padding:20px 0; font-family:Georgia, "Times New Roman", Times, serif;}
```

```
#header #logo{display:block; float:left; width:630px;}
```

```
#header #logo h1{border:none; margin:0; padding:0; float:left;}
```

```
#header #logo p{display:block; float:left; height:25px; margin:0 0 0 10px; padding:15px 0 0 10px; border-left:2px solid #39669D;}
```

```
#header #logo h1 a{font-size:36px; color:#FFFFFF; background-color:#1F3D63;}
```

```
#header #info{display:block; float:right; margin-top:15px;}
```

```
#header #info ul{margin:0; padding:0; list-style:none;}
```

```
#header #info li{display:inline; margin:0 0 0 10px;}
```

```
/* .....Topbar..... */
```

```
#topbar{z-index:1000;}
```

```
#topbar #search{display:block; float:right; width:243px; margin:10px 0 0 0; padding:0;}
```



## **10.RESULT**

### **10.1 SYSTEM TEST**

#### **Testing**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

#### **TESTING OBJECTIVES**

To ensure that during operation the system will perform as per specification.

TO make sure that system meets the user requirements during operation

To make sure that during the operation, incorrect input, processing and output will be detected

To see that when correct inputs are fed to the system the outputs are correct

To verify that the controls incorporated in the same system as intended

Testing is a process of executing a program with the intent of finding an error

A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.,

### 10.1.1 TEST CASE DESIGN:

#### **White box testing**

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independent paths in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

#### **Black Box Testing**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

##### **Unit Testing:**

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is to test the internal logic of the module/program. In the Generic code project, the unit testing is done during the coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

##### **Integration Testing:**

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis is on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

##### **Validation Testing**

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also

aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

### **System Testing**

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

Implementing the system in a simulated production environment and testing it.

Introducing errors and testing for error handling.

### **TEST CASES:**

#### **Test case1:**

Test case for Login form:

<b>FUNCTION:</b>	<b>LOGIN</b>
<b>EXPECTED RESULTS:</b>	<b>Should Validate the user and check his existence in database</b>
<b>ACTUAL RESULTS:</b>	<b>Validate the user and checking the user against the database</b>
<b>LOW PRIORITY</b>	<b>No</b>
<b>HIGH PRIORITY</b>	<b>Yes</b>

### Test case2:

Test case for User Registration form:

FUNCTION:	USER REGISTRATION
EXPECTED RESULTS:	Should check if all the fields are filled by the user and saving the user to database.
ACTUAL RESULTS:	Checking whether all the fields are field by user or not through validations and saving user.
LOW PRIORITY	No
HIGH PRIORITY	Yes

### Test case3:

Test case for Change Password:

When the old password does not match with the new password , then this results in displaying an error message as “ OLD PASSWORD DOES NOT MATCH WITH THE NEW PASSWORD”.

### Test case 4:

Test case for Forget Password:

When a user forgets his password he is asked to enter Login name, ZIP code, Mobile number. If these are matched with the already stored ones then user will get his Original password.

Module	Functionality	Test Case	Expected Results	Actual Results	Result	Priority
--------	---------------	-----------	------------------	----------------	--------	----------

User	Login Usecase	1. Navigate To Www.Sample.Co m  2. Click On Submit Button Without Entering Username and Password	A Validation Should Be As Below “Please Enter Valid Username & Password”	A Validation Has Been Populated As Expected	Pass	High
		1. aNavigate To Www.Sample.Co m  2. Click On Submit Button With Out Filling Password And With Valid Username	A Validation Should Be As Below “Please Enter Valid Password Or Password Field Can Not Be Empty “	A Validation Is Shown As Expected	Pass	High
		1. NNavigate To Www.Sample.Co m  2. Enter Both Username And Password Wrong And Hit Enter	A Validation Shown As Below “The Username Entered Is Wrong”	A Validation Is Shown As Expected	Pass	High

		1. Navigate To Www.Sample.Co m  2. Enter Validate Username And Password And Click On Submit	Validate Username And Password In DataBase And Once If They Correct Then Show The Main Page	Main Page/ Home Page Has Been Displayed	Pass	High
--	--	---	--	--	------	------

## 10.2 OUTPUT SCREENS

### Home screen



## Admin login

Detection of Cyber Harassers

on Social Media

HomeAdminUserRegistration



### About Project

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. • Desired Solution : 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

ADMIN LOGIN

UserName

Password

Login

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

## Admin login status

localhost:8084 says  
Login Successful...!!

OK

Activate Windows  
Go to Settings to activate Windows.



## Admin home

Detection of Cyber Harassers

on Social Media

Home

Add Category

Add Words

Cyber Harasser

Logout



### About Project

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. • Desired Solution : 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

WELCOME TO ADMIN HOME

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

## Add category

Detection of Cyber Harassers

on Social Media

Home

Add Category

Add Words

Cyber Harasser

Logout

### ADD CATEGORY

category

Add Category

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

add all categories

ADD CATEGORY WORDS

Select Category	<input type="text"/>
Enter Word	<input type="text"/>
	<input type="button" value="Add Word"/>

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

Cyber harassers

CYBER HARASSARS

Harassar Name	Post Content
venkat	enmity
kishan	arnab gosvami got Assaulted by mombay police

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

## User login

Detection of Cyber Harassers

on Social Media

HomeAdminUserRegistration



### About Project

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. • Desired Solution : 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

USER LOGIN

UserName

Password

LoginRegister

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

## User registrations

Detection of Cyber Harassers

on Social Media

HomeAdminUserRegistration



### About Project

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. • Desired Solution : 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

USER REGISTRATION

Name

Email

Mobile

Address

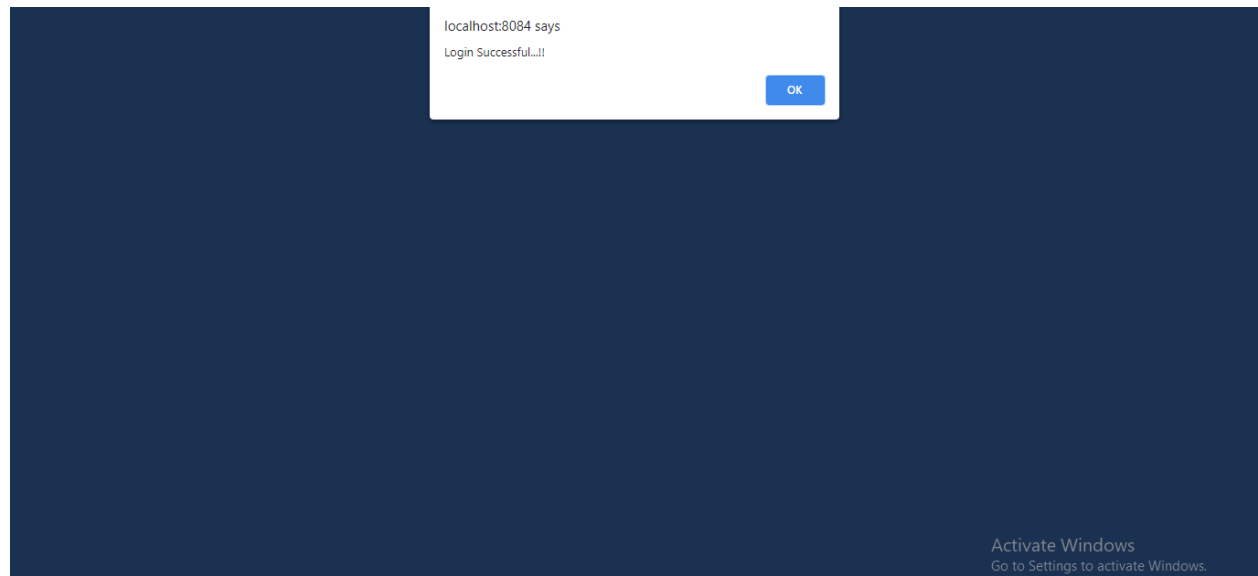
UserName

Password

RegisterLogin

Activate Windows  
Go to Settings to activate Windows.

## User login status



## User home



## Post content

Detection of Cyber Harassers

on Social Media

[Home](#)[Post Content](#)[View All Posts](#)[Logout](#)

POST CONTENT TO PUBLIC

Title

Content

Image

Choose File

No file chosen

Title

Post Content

Detection of Cyber Harassers on Social Media

Activate Windows  
Go to Settings to activate Windows.

## view all posts

Detection of Cyber Harassers

on Social Media

[Home](#)[Post Content](#)[View All Posts](#)[Logout](#)

ALL POSTS

Title

Arbab Arrest

Image

Content

arnab gosvami got Assaulted by mombay police

Comment

[Click](#)

Title

null

Image

Content

enmity

Comment

[Click](#)

Activate Windows  
Go to Settings to activate Windows.

## **11.CONCLUSION**

As each and every one even child is using internet nowadays and getting harassed by predators so in order to stop these predators it is very important to detect and punish them. The main aim of the groomer is to build a relationship with a child in order to gain access to that child. When grooming takes place, it is common that an adult groomer is pretending to be a child with common hobbies or interests to build a relationship with child. In this project we detect child predator for child safety. And send predator report to cyber admin for action.

### **11.1 FUTURE SCOPE**

The future scope of detecting and reporting online child predators is vast and critical for ensuring the safety of children in the digital age. With advancements in artificial intelligence and machine learning, the development of sophisticated algorithms can significantly enhance the accuracy and speed of identifying potential predators. These technologies can be integrated into social media platforms, messaging apps, and online gaming environments to monitor and analyze interactions in real-time, ensuring swift intervention when suspicious activities are detected. Additionally, the implementation of blockchain technology can secure and anonymize reports, protecting the privacy of both victims and whistleblowers while ensuring that data remains tamper-proof. As more devices become interconnected through the Internet of Things (IoT), these detection systems can expand to various digital touchpoints, providing a comprehensive safety net for children across different online platforms.

## 12.REFERENCES

1. Fauzi, M. A., & Bours, P. (2020, April). Ensemble Method for Sexual Predators Identification in Online Chats. In 2020 8th International Workshop on Biometrics and Forensics (IWBF) (pp. 1-6). IEEE.
2. Ashcroft, M., Kaati, L., & Meyer, M. (2015, September). A Step Towards Detecting Online Grooming--Identifying Adults Pretending to be Children. In 2015 European Intelligence and Security Informatics Conference (pp. 98-104). IEEE.
3. Altay, E. V., & Alatas, B. (2018, December). Detection of cyberbullying in social networks using machine learning methods. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 87-91).
4. Cano, A. E., Fernandez, M., & Alani, H. (2014, November). Detecting child grooming behaviour patterns on social media. In International conference on social informatics (pp. 412-427). Springer, Cham.
5. Haidar, B., Chamoun, M., & Yamout, F. (2016, November). Cyberbullying detection: A survey on multilingual techniques. In 2016 European Modelling Symposium (EMS) (pp. 165-171). IEEE.
6. Rodríguez, J. I., Durán, S. R., Díaz-López, D., Pastor-Galindo, J., & Mármol, F. G. (2020). C3-sex: A conversational agent to detect online sex offenders. *Electronics*, 9(11), 1779.
7. Dombrowski, S. C., LeMasney, J. W., Ahia, C. E., & Dickson, S. A. (2004). Protecting children from online sexual predators: technological, psychoeducational, and legal considerations. *Professional Psychology: Research and Practice*, 35(1), 65.
8. Wolak, J., Finkelhor, D., Mitchell, K. J., & Ybarra, M. L. (2010). Online “predators” and their victims: Myths, realities, and implications for prevention and treatment.
9. Cano, A. E., Fernandez, M., & Alani, H. (2014, November). Detecting child grooming behaviour patterns on social media. In International conference on social informatics (pp. 412-427). Springer, Cham.