A Major Project Report on

DETECTING MALICIOUS SOCIAL BOTS BASED ON CLICKSTREAM SEQUENCES

Submitted to the

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

In partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

RAVI RAMYA SMRUTHI - 16WJ1A05R0

SABHAVATH ASHWINI - 16WJ1A05T0

NENAVATH SUSHMITHA - 16WJ1A05L1

Under the Esteemed Guidance Of

Mrs. Padma Rajani

Assistant Professor, CSE Dept.

GURU NANAK INSTITUTIONS TECHNICAL CAMPUS (AUTONOMOUS)

School of Engineering and Technology

Ibrahimpatnam, R.R District 501506

2019-2020

GNI
www.gniindia.org

GURU NANAK INSTITUTIONS
TECHNICAL CAMPUS

Approved by AICTE - New Delhi | Affiliated to JNTU - Hyderabad | Accredited by National Assessment and Accreditation Council | Accredited by National Board of Accreditation

AUTONOMOUS
under Section 2 (f) & 12 (b) of
University Grants Commission Act

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that this project report entitled "**DETECTING MALICIOUS SOCIAL BOTS BASED ON CLICKSTREAM SEQUENCES**" by **RAVI RAMYA SMRUTHI (16WJ1A05R0), SABHAVATH ASHWINI (16WJ1A05T0), NENAVATH SUSHMITHA (16WJ1A05L1)** submitted in partial fulfilment the requirements for the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the **Jawaharlal Nehru Technological University, Hyderabad** during the academic year 2019-2020, is a bonafide record of work carried out under our guidance and supervision.

INTERNAL GUIDE	**PROJECT COORDINATOR**	**HOD CSE2**
Mrs. Padma Rajani	**Mrs. V. Swathi**	**Mr. V. Devasekhar**

EXTERNAL EXAMINER

Ostilio®

PROJECT COMPLETION CERTIFICATE

This is to certify that the following students of final year B.Tech, Department of ***Computer Science and Engineering*** *Guru Nanak Institutions Technical Campus (GNITC), have completed their training and project at GNITC successfully.*

Student Name	**Roll Number**
RAVI RAMYA SMRUTHI	16WJ1A05R0
SABHAVATH ASHWINI	16WJ1A05T0
NENAVATH SUSHMITHA	16WJ1A05L1

The training was conducted on **RED HAT® Cloud Computing** in Hybrid Cloud Computing In-Campus Lab with RED HAT Academy at GNI Campus with the project titled:

"DETECTING MALICIOUS SOCIAL BOTS BASED ON CLICKSTREAM SEQUENCES" During Academic Year 2019 – 2020.

Authorized Signature

Ostilio Systems Private Limited

[illegible]

D.NO. 3-6-290, Block A, #306,
Third Floor, Sai Tirumala Towers,
Hyderguda, Hyderabad - 500029

ACKNOWLEDGEMENT

We wish to express our sincere thanks to **Dr. Rishi Sayal**, Associate Director, GNITC for providing us the conductive environment for carrying through our academic schedule and project with ease.

We have been truly blessed to have a wonderful adviser, **Mr. V Devasekhar**, HOD of CSE, GNITC for guiding us to explore the ramification of our work and we express my sincere gratitude towards him for leading me thought the completion of project.

We thank Project Coordinator, **Mrs. V Swathi,** Associate Professor, CSE, GNITC for providing seamless support and right suggestions that are given in the development of the project.

We especially thank our internal guide, **Mrs. Padma Rajani,** Assistant Professor, CSE, GNITC, for her constant guidance in every stage of the project. We would also like to thank all our lecturers for helping us in every possible way whenever the need arose.

On a more personal note we thank our beloved parents and friends for their moral support during our project.

RAVI RAMYA SMRUTHI - 16WJ1A05R0

SABHAVATH ASHWINI - 16WJ1A05T0

NENAVATH SUSHMITHA - 16WJ1A05L1

ABSTRACT

Social Media and Online Social Networks are having a deep impact on our daily life, giving voice to the crowd, and reshaping the information. There have been various methods of collecting and analysing Big Data, as there is a significant development in the volume, velocity, and variety of user-generated data in Social Networking Websites.

Social bots are a growing presence and problem on social media. A social bot is an agent that communicates autonomously on social media, often with the task of influencing the course of discussion and/or the opinions of its readers. A social bot is basically, programmed to send automated responses to the user and perform related operations based on few procedures. Anyhow, there are malicious social bots which spread wrong information (e.g., fake news), that can affect the cyber-world. In order that these social bots do not create a havoc and result in real-time consequences, we must detect and remove such kind of harmful social bots. The difference between a normal user and a social bot must be known, as the social bots are more intelligent and can mimic user behaviour with ease, which becomes difficult in detecting the malicious social bots. In order to differentiate between social bots and normal users, detecting malicious social bots, and minimize the loss incurred by them, we must gather and analyse the user behaviour and recognize the difference between malicious social bots and normal users dynamically.

Normal Users are distinguished from Social Bots based on six features i.e., Network, User, Making friends, Time, Content and Emotion. This approach uses Naive Bayes Algorithm (supervised learning algorithm) to categorize Twitter accounts into normal users and social bots. However, in supervised learning algorithms we must define and train a large amount of data, which requires much time, manpower and is generally not applicable for Big Data Social Networking Environment. In order to detect the malicious social bots better in online social networks, we use semi-supervised learning algorithm. We analyse the features of user behaviour and recognize the user Clickstreams based on Transition Probability. Clickstream is a record of user's activity on the internet, which tracks every website and every page visited by the user, the time spent on every page and in what order the pages were visited. The Transition Probability is the probability of transitioning from one state to another in a single step. This method not only analyses the Transition probability of user's Clickstream but also includes the time duration of their behaviour.

***Keywords*:** Online Social Network, social bots, user behaviour, transition probability, Clickstreams.

TABLE OF CONTENTS

Content **Page no.**

LIST OF FIGURES

1. PROJECT DESCRIPTION

1.1 INTRODUCTION

In online social networks, social bots are social accounts controlled by automated programs that can perform corresponding operations based on a set of procedures. The increasing use of mobile devices (e.g., Android and iOS devices) also contributed to an increase in the frequency and nature of user interaction via social networks. It is evidenced by the significant volume, velocity and variety of data generated from the large online social network user base. Social bots have been widely deployed to enhance the quality and efficiency of collecting and analysing data from social network services. For example, the social bot SF QuakeBot is designed to generate earthquake reports in the San Francisco Bay, and it can analyse earthquake related information in social networks in real-time. However, public opinion about social networks and massive user data can also be mined or disseminated for malicious or nefarious purpose.

In online social networks, automatic social bots cannot represent the real desires and intentions of normal human beings, so they are usually looked upon malicious ones. For example, some fake social bots accounts created to imitate the profile of a normal user, steal user data and compromise their privacy, disseminate malicious or fake information, malicious comment, promote or advance certain political or ideology agenda and propaganda, and influence the stock market and other societal and economical markets. Such activities can adversely impact the security and stability of social networking platforms. In previous research, various methods were used to protect the security of online social network. User behaviour is the most direct manifestation of user intent, as different users have different habits, preferences, and online behaviour (e.g., the way one clicks or types, as well as the speed of typing). In other words, we may be able to mine and analyse information hidden in user's online behaviour to problem and identify different users. However, we also need to be conscious of situational factors that may play a role in changing user's online behaviour. In other words, user behaviour is dynamic, and its environment is constantly changing _ i.e., external observable environment (e.g., environment and behaviour) of application context and the hidden environment in user information. In order to distinguish social bots from normal users accurately, detect malicious social bots, and reduce the harm of malicious social bots, we need to acquire and analyse social situation of user behaviour and compare and understand the differences of malicious social bots and normal users in dynamic behaviour.

Specifically, in this paper, we aim to detect malicious social bots on social network platforms in real-time, by

i. proposing the transition probability features between user clickstreams based on the social situation analytics; and
ii. designing an algorithm for detecting malicious social bots based on spatiotemporal features.

1.2 OBJECTIVE OF THE PROJECT

Fake accounts amplify social signals including likes, shares, views, and skew the results of online polls. Bots distort the data that is used for opinion mining, social listening, and consumer behavior prediction. Botnet operators perform credential stuffing attacks and use stolen credentials to take over accounts. Compromised accounts of businesses and individuals lead to unauthorized publication of confidential information, reputational damage, financial losses, and disclosure of personal data.

We collect device, browser, and account-specific parameters to differentiate between genuine and fake accounts. We provide detailed traffic analytics based on the analysis of user interactions to help you filter fake accounts and stop illegal activities through these accounts.

1.3 METHODOLOGY

This project contains 2 modules:

1. **OSN Server**: In this module, the OSN Server has to login by using valid username and password. After login successful he can do some operations such as view all user details and authorize them, list of all friends requests and response ,View all posts like images and messages user, view all Similar group users like doctors, Engineers, Business Man, etc.,. OSN Server can add some BOTS words to the database and view all words added by him and based on that negative words admin can find all users behavior and produce chart for that behavior words.
    - View and Authorize Users
    - In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, username, email, address and admin authorize the users.
2. **User**: In this module, there are n numbers of users are present. User should register with group option before doing some operations. After registration successful he has to wait for admin to authorize him and after admin authorized user can login by using authorized username and password. Login successful he will do some operations like view profile details, Search friends

based on keyword or friends name, view the friend requests, post message with image to all friends. Find posts of friends and comment on that posts.

- Can also view all his friends and delete those who don't want, view all group posts like doctor or engineer etc.
- Viewing Profile Details
- The user can see their own profile details, such as their address, email, mobile number, profile Image
- Search Friends, Request, and View Friend Requests, View all Friend Details
- Search for other users by their names, send requests and view friend requests from other users. User can see all his friend details with their images and personnel details

2. SYSTEM ANALYSIS

2.1 PROBLEM ANALYSIS

In online social networks, automatic social bots cannot represent the real desires and intentions of normal human beings, so they are usually looked upon malicious ones. For example, some fake social bots accounts created to imitate the profile of a normal user, steal user data and compromise their privacy, disseminate malicious or fake information, malicious comment, promote or advance certain political or ideology agenda and propaganda, and influence the stock market and other societal and economical markets. Such activities can adversely impact the security and stability of social networking platforms.

2.2 EXISTING SYSTEM

In existing system, social bots have been used to manage automated analytical services and provide better quality services to customers. However, malicious social bots are also used to spread false information, and this can lead to real-world consequences. Therefore, it is important to identify and eliminate malicious social bots on online social networks. Most current detection methods of malicious social bots analyze quantitative characteristics of their behavior. These bots can easily mimic social bots. Thereby resulting in less accuracy of analysis.

Disadvantages

- Malicious social bots are also used to spread false information, and this can lead to real world consequences
- Less accuracy.

2.3 PROPOSED SYSTEM

In this proposed system, most current detection methods of malicious social bots analyze quantitative characteristics of their behavior. These bots can easily mimic social bots; Thereby resulting in less accuracy of analysis. A novel method for detecting malicious social bots is presented in this project, including the conversion probability of clickstream sequences and the selection of two features based on semi-monitored clustering. This method not only analyzes the transformation probability of user behavior clickstreams, but also considers the behavior of the timestamp.

Advantages

- This method analyzes the conversion probability of user behavior clickstreams.
- Used to detect malicious social bots on social network platforms in real-time.

2.4 PROJECT MODULES

This project contains 2 modules:

1. **OSN Server**: In this module, the OSN Server has to login by using valid username and password. After login successful he can do some operations such as view all user details and authorize them, list of all friends requests and response ,View all posts like images and messages user, view all Similar group users like doctors, Engineers, Business Man, etc.,. OSN Server can add some BOTS words to the database and view all words added by him and based on that negative words admin can find all users behavior and produce chart for that behavior words.
   - View and Authorize Users
   - Can view the list of users who all registered. In this, the admin can view the user’s details such as, username, email, address and admin authorize the users.
2. **User**: In this module, there are n numbers of users are present. User should register with group option before doing some operations. After registration successful he has to wait for admin to authorize him and after admin authorized user can login by using authorized username and password. Login successful he will do some operations like view profile details, Search friends based on keyword or friends name, view the friend requests, post message with image to all friends. Find posts of friends and comment on that posts.
   - Can also view all his friends and delete those who don’t want, view all group posts like doctor or engineer etc.
   - Viewing Profile Details
   - Can see their own profile details, such as their address, email, mobile number, profile Image
   - Search Friends, Request, and View Friend Requests, View all Friend Details
   - Search for other users by their names, send requests and view friend requests from other users
   - Can see all his friend details with their images and personnel details

2.5 LITERATURE SURVEY

Reviewing the literature is an essential component when writing academic papers that use research findings for ideas and points, they try to make. It is also a requirement for the project report. A **literature survey** in a project report represents the study done to assist in the completion of a project. A literature survey also describes a survey of the previous existing material on a topic of the report. A project report is an assessment during a process or project conveying these details:

- Accomplished sub goals
- Expended resources
- Problems encountered
- Estimation on whether the project is likely to complete on time and within provided budget

The focus of a literature survey in on the following and in this order:

- Existing theories on a topic with universal acceptance across the board
- Books on the subject acting as a reference for the concepts that project uses whether they are specific or generic.
- Research papers might be a reference for theories nut most cases require a critical comparison to establish the purpose of the project and improvement
- You may also include another project report and what helped you
- Challenges for the project and by ongoing work if it is available

Case Study 1:

Title : Detecting Malicious Social Bots based on Clickstream Sequences

Authors : Peining Shi, Zhiyong Zhang, Kim-Kwang Raymond Choo

Year : 2019

Abstract : With the significant increase in the volume, velocity and variety of user data (e.g., user-generated data) in online social networks, there have been attempted to design new ways of collecting and analysing such big data. For example, social bots have been used to perform automated analytical services and provide users with improved quality of service. However, malicious social bots have also been used to disseminate false information (e.g., fake news), and this can result in real-world consequences. Therefore, detecting and removing malicious social bots in online social networks is crucial. Most existing detection methods of malicious social bots analyse the quantitative features of their behaviour. These features are easily imitated by social bots, thereby resulting in low accuracy of analysis. A novel method of detecting malicious social bots, including both features selection based on the transition probability of clickstream sequences and semi-supervised clustering is presented in this paper. This method not only analyses transition probability of user behaviour clickstreams, but also considers the time feature of behaviour. Findings from our experiments on real online social network platforms demonstrate that detection accuracy for different types of malicious social bots by the detection method of malicious social bots based on transition probability of user behaviour clickstreams increases by an average of 12.8%, in comparison to the detection method based on quantitative analysis of user behaviour.

Case Study 2:

<u>Title</u> : Unsupervised Clickstream Clustering for User Behaviour Analysis

<u>Authors</u> : Gang Wang, Xinyi Zhang, Shiliang Tang, Haitao Zheng, Ben Y. Zhao

<u>Year</u> : 2016

<u>Abstract</u> : Online services are increasingly dependent on user participation. Whether it's online social networks or crowdsourcing services, understanding user behaviour is important yet challenging. In this paper, we build an unsupervised system to capture dominating user behaviours from clickstream data (traces of users' click events) and visualize the detected behaviours in an intuitive manner. Our system identifies "clusters" of similar users by partitioning a similarity graph (nodes are users; edges are weighted by clickstream similarity). The partitioning process leverages iterative feature pruning to capture the natural hierarchy within user clusters and produce intuitive features for visualizing and understanding captured user behaviours. For evaluation, we present case studies on two large-scale clickstream traces (142 million events) from real social networks. Our system effectively identifies previously unknown behaviours, e.g., dormant users, hostile chatters. Also, our user study shows people can easily interpret identified behaviours using our visualization tool.

Case Study 3:

Title : BotOrNot: A System to Evaluate Social Bots

Authors : Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, Filippo Menczer

Year : 2016

Abstract : While most online social media accounts are controlled by humans, these platforms also host automated agents called social bots or sybil accounts. Recent literature reported on cases of social bots imitating humans to manipulate discussions, alter the popularity of users, pollute content and spread misinformation, and even perform terrorist propaganda and recruitment actions. Here we present BotOrNot, a publicly available service that leverages more than one thousand features to evaluate the extent to which a Twitter account exhibits similarity to the known characteristics of social bots. Since its release in May 2014, BotOrNot has served over one million requests via our website and APIs.

3. SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 REQUIREMENT ANALYSIS

A Subject People are Interested In: The most successful blogs are written about topics that have a broad appeal. The more people who are interested in the topic you write about at your blog, the more people will search for information on that topic and want to read about it.

Passion for Your Subject: As a blogger, you must write about your blog's subject a lot. The writing is nonstop. If you don't love your blog's topic, it will show. Readers can detect when a blogger is just going through the motions rather than speaking from the heart.

Commitment: Successful blogging requires a massive amount of sweat equity and dedication. Building a successful blog requires more than just publishing a new post a few times a week. The most successful blogs are updated frequently (often several times each day), and the bloggers behind those blogs work relentlessly to promote their blogs and drive traffic to them.

Time: Building a successful blog requires a massive time investment. Growing a blog doesn't stop with publishing post. Top bloggers spend a lot of time everyday promoting their blogs, researching, and reading to drive traffic to it.

A Desire to Network: Socializing is a critical component of developing a successful blog. Blogging by nature is a social medium, and successful blogs become so primarily because of the strong sense of community surrounding them. Top bloggers take time to respond to comments and interact with their blogs' readers as well as network on social sites, forums and more all to further promote their blogs.

A Desire to Keep Learning: The blogosphere is ever-changing, which means top bloggers are always looking for new ways to enhance their blogs by persistently researching anything and everything related to blogging.

A Love of Research and Reading: Successful bloggers read a lot in order to stay updated on their blogs' topics and the blogosphere.

Creativity: A successful blog is typically updated very frequently. That means successful blogs always provide fresh, unique content to their readers. Writing that new content requires a great deal of creativity to keep readers from feeling bored or from feeling like the blog does not bring them any value.

Patience: Blogging success does not happen overnight. Be prepared to stay dedicated to promoting your blog for the long haul. Eventually, your hard work should pay off with increased traffic and a significant growth in popularity of your blog. The key is to not give up.

Hardware Requirements:

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.

Software Requirements:

- Back End: JDBC Connection
- Operating System: Windows 10/8/7/Vista/XP
- Scripting Language: JavaScript
- Style Sheets: CSS

3.2 TECHNOLOGIES USED IN THE PROJECT

3.2.1 JavaScript

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offers proven portability, open standards, and a mature re-usable component model. The Java Server Pages architecture enables the separation of content generation from content presentation. This separation does not ease maintenance headaches, it also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other's work.

Features of JSP:

Portability:

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

Components:

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components current supported include Java Beans, and Servlets.

Processing:

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

Access Models:

A Java Server Pages file may be accessed in at least two different ways. A client's request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that perform well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

Steps in the execution of a JSP Application:

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server gives this request to the JSP engine.
3. JSP engine is program which can understands the tags of the jsp and then it converts those tags into a Servlet program, and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the result is given back to the JavaWebServer and then it is transferred back to the client.

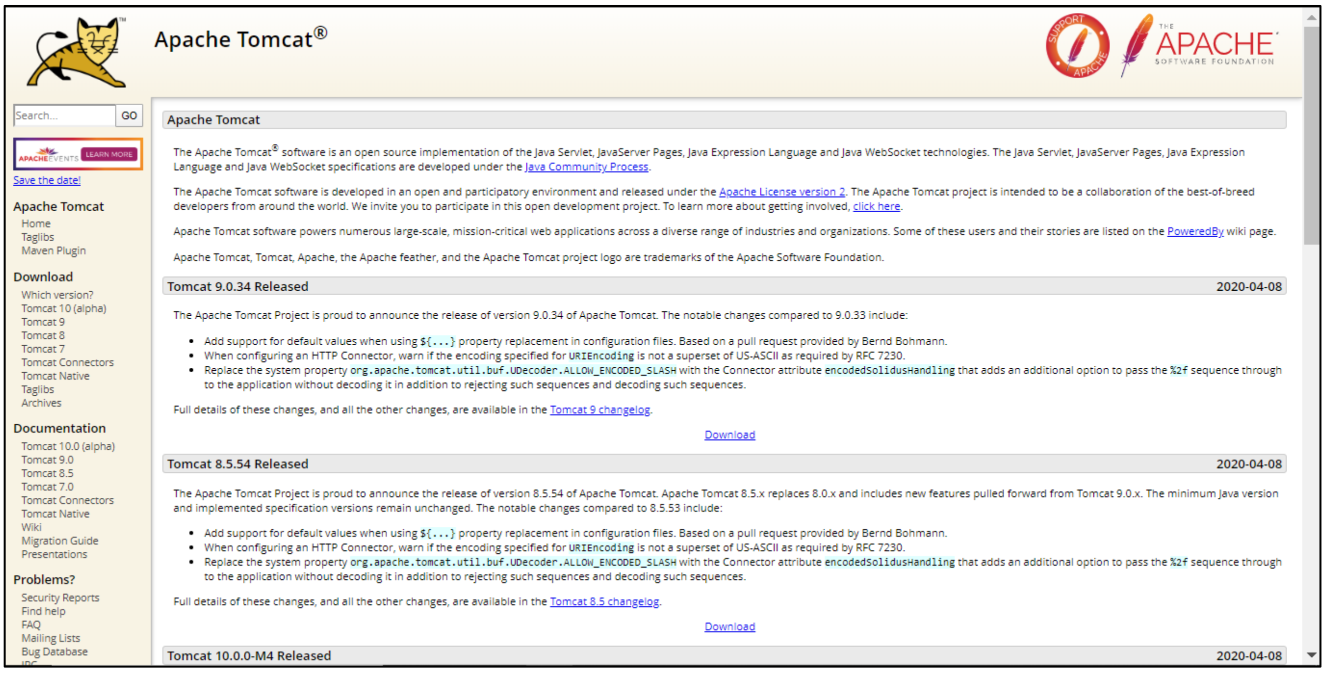
JDBC connectivity:

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

- Perform connection and authentication to a database server
- Manager transactions
- Move SQL statements to a database engine for preprocessing and execution
- Execute stored procedures
- Inspect and modify the results from Select statements.

3.2.2 Tomcat Web Server

Tomcat is an open source web server developed by Apache Group. Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed by Sun under the Java Community Process. Web Servers like Apache Tomcat support only web components while an application server supports web components as well as business components (BEAs WebLogic, is one of the popular application server).To develop a web application with jsp/servlet install any web server like JRun, Tomcat etc. to run your application.



Apache Tomcat

The Apache Tomcat® software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process.

The Apache Tomcat software is developed in an open and participatory environment and released under the Apache License version 2. The Apache Tomcat project is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, click here.

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the PoweredBy wiki page.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation.

Tomcat 9.0.34 Released 2020-04-08

The Apache Tomcat Project is proud to announce the release of version 9.0.34 of Apache Tomcat. The notable changes compared to 9.0.33 include:

- Add support for default values when using `${...}` property replacement in configuration files. Based on a pull request provided by Bernd Bohmann.
- When configuring an HTTP Connector, warn if the encoding specified for `URIEncoding` is not a superset of US-ASCII as required by RFC 7230.
- Replace the system property `org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH` with the Connector attribute `encodedSolidusHandling` that adds an additional option to pass the `%2f` sequence through to the application without decoding it in addition to rejecting such sequences and decoding such sequences.

Full details of these changes, and all the other changes, are available in the Tomcat 9 changelog.

Download

Tomcat 8.5.54 Released 2020-04-08

The Apache Tomcat Project is proud to announce the release of version 8.5.54 of Apache Tomcat. Apache Tomcat 8.5.x replaces 8.0.x and includes new features pulled forward from Tomcat 9.0.x. The minimum Java version and implemented specification versions remain unchanged. The notable changes compared to 8.5.53 include:

- Add support for default values when using `${...}` property replacement in configuration files. Based on a pull request provided by Bernd Bohmann.
- When configuring an HTTP Connector, warn if the encoding specified for `URIEncoding` is not a superset of US-ASCII as required by RFC 7230.
- Replace the system property `org.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH` with the Connector attribute `encodedSolidusHandling` that adds an additional option to pass the `%2f` sequence through to the application without decoding it in addition to rejecting such sequences and decoding such sequences.

Full details of these changes, and all the other changes, are available in the Tomcat 8.5 changelog.

Download

Tomcat 10.0.0-M4 Released 2020-04-08

Fig 3.2.2.1 Tomcat Web Server

3.2.3 Cloud Computing using AWS

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe datacenters available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings. AWS services can offer an organization tools such as compute power, database storage and content delivery services.

AWS is separated into different services and each can be configured in different ways based on the user's needs. Users should be able to see configuration options and individual server maps for an AWS service. More than 100 services comprise the Amazon Web Services portfolio, including those for compute, databases, infrastructure management, application development and security.

These services, by category, include:

- Compute
- Storage databases
- Data management
- Migration
- Hybrid cloud
- Networking
- Development tools
- Management
- Monitoring
- Security
- Governance
- Big data management
- Analytics
- Artificial intelligence (AI)
- Mobile development
- Messages and notification

3.3 SDLC (Software Development Life Cycle)

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

This term is also known as the software development process.

SDLC consists of following activities:

Planning: The most important parts of software development, requirement gathering, or requirement analysis are usually done by the most skilled and experienced software engineers in the organization. After the requirements are gathered from the client, a scope document is created in which the scope of the project is determined and documented.

Implementation: The software engineers start writing the code according to the client's requirements.

Testing: This is the process of finding defects or bugs in the created software.

Documentation: Every step in the project is documented for future reference and for the improvement of the software in the development process. The design documentation may include writing the application programming interface (API).

Maintaining: Software maintenance is done for future reference. Software improvement and new requirements (change requests) can take longer than the time needed to create the initial development of the software.

There are several software development models followed by various organizations:

Waterfall Model: This model involves finishing the first phase completely before commencing the next one. When each phase is completed successfully, it is reviewed to see if the project is on track and whether it is feasible to continue.

V-Shaped Model: This model focuses on execution of processes in a sequential manner, like the waterfall model but with more importance placed on testing. Testing procedures are written even before the commencement of writing code. A system plan is generated before starting the development phase.

Incremental Model: This life cycle model involves multiple development cycles. The cycles are divided up into smaller iterations. These iterations can be easily managed and go through a set of phases including requirements, design, implementation, and testing. A working version of the software is produced during the first iteration, so working software is created early in the development process.

4. SYSTEM DESIGN

4.1 INTRODUCTION

Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specifies requirements. System design could be as seen as the application of systems theory to product development. It is about the physical organization of the system. It is demonstrated with the help of UML diagram or block diagrams etc. it is explained in a pictorial representation.

4.2 SYSTEM ARCHITECTURE

The architecture depicts the mechanism of detecting the malicious social bots based on a category of using negative words. There are 3 elements i.e., Web Database, OSN Server and User. The project uses unsupervised learning algorithm named, “Transition Probability” to detect the malicious social bots which may be responsible for spreading wrong information.

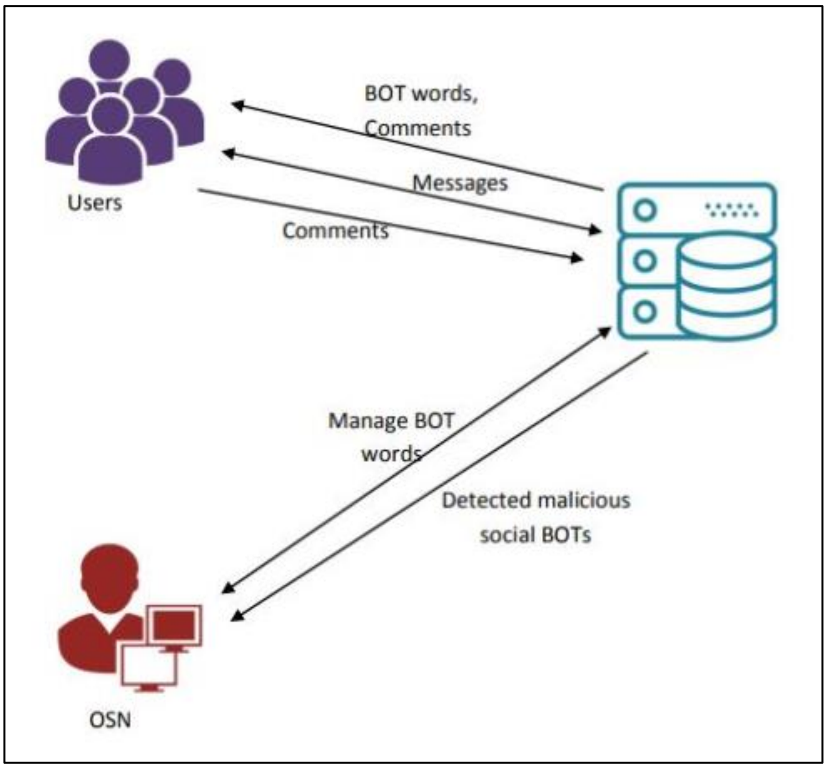


Fig 4.2.1: Architecture for Detecting Malicious Social Bots using Clickstream Sequences

- **OSN Server**: The server trusted agent i.e., the Administrator manages the OSN Server. The admin is responsible for viewing, authorizing, and managing all user’s profiles and their posts. The admin can track the user behaviour using Clickstream Sequences and prepares a behaviour chart based on their activities. He adds the BOT words which are not supposed to be used and detects the users who post or comment using the specified BOT words. These detected users are listed under Malicious Social Bots by the administrator. The admin is responsible for replying to the user queries and make sure the user does not have any problem. The BOT words which are added by the admin is stored in the Database.

- **User**: The user must register under a group of his choice and login. He can check his profile and can edit it. He can send requests to his friends and receive requests from other friends. He can view and comment on his friend's posts. The group posts can be viewed by the user. He can delete the friends he does not want. He can post messages with images.
- **Web Database**: The BOT words, user details, admin details, user behaviour, user queries, admin responses and user friends are stored in the Database.

4.3 UML DIAGRAMS

The Unified Modelling Language (UML) is a general-purpose modelling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system. The Unified Modelling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram.

UML is **not a programming language**; it is rather a visual language. We use UML diagrams to portray the **behavior and structure** of a system. UML helps software engineers, businessmen and system architects with modelling, design, and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

Diagrams can be viewed as:

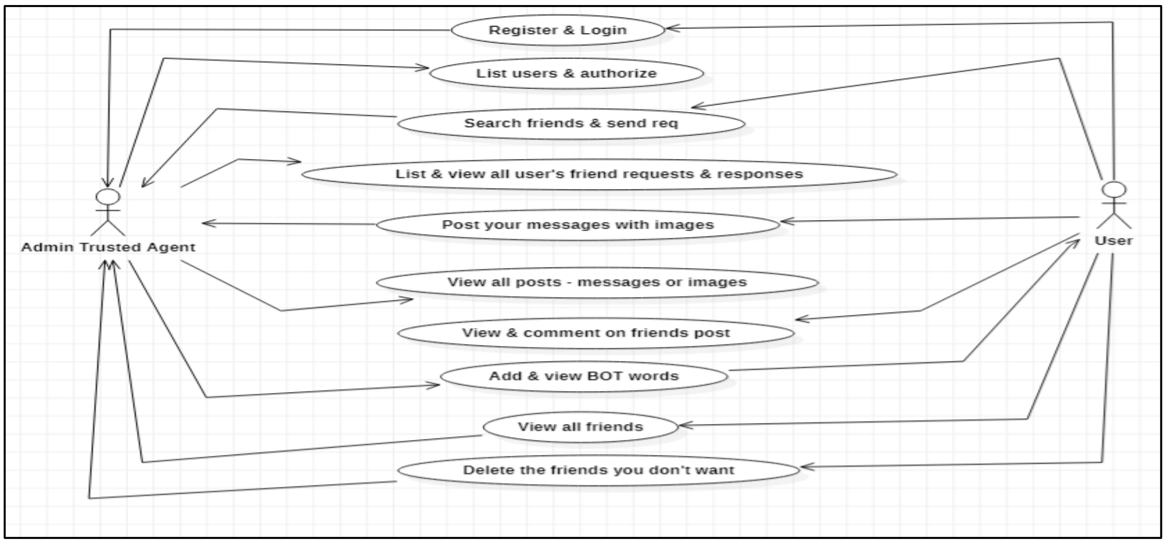
- Structural Diagrams:
    - Class Diagram
    - Composite Structure Diagram
    - Object Diagram
    - Component Diagram

- Deployment Diagram
- Package Diagram

- Behaviour Diagrams:
  - State Machine Diagram
  - Activity Diagram
  - Use Case Diagram
  - Sequence Diagram
  - Communication Diagram
  - Timing Diagram
  - Interaction Overview Diagram

4.3.1 Use Case Diagram for Detecting Malicious Social Bots using Clickstream Sequences:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.



Register & Login
List users & authorize
Search friends & send req
List & view all user's friend requests & responses
Admin Trusted Agent
Post your messages with images
User
View all posts - messages or images
View & comment on friends post
Add & view BOT words
View all friends
Delete the friends you don't want

Fig 4.3.1: Use Case Diagram for Detecting Malicious Social Bots using Clickstream Sequences

4.3.2 Class Diagram for Detecting Malicious Social Bots using Clickstream Sequences :

A class diagram in the Unified Modelling 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 objects.

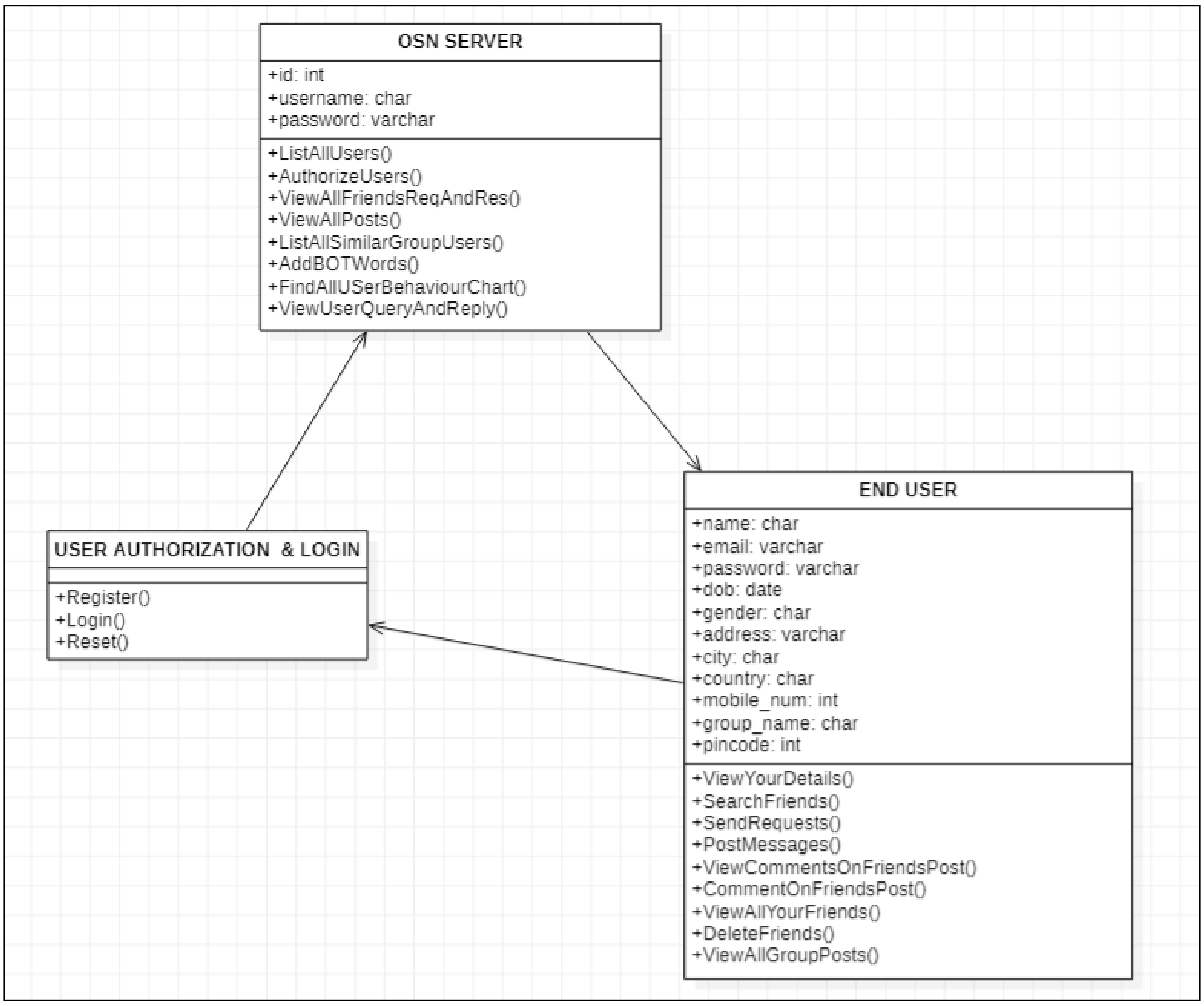


Fig 4.3.2: Class Diagram for Detecting Malicious Social Bots using Clickstream Sequences

4.3.3 Sequence Diagram for Detecting Malicious Social Bots using Clickstream Sequences:

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios.**

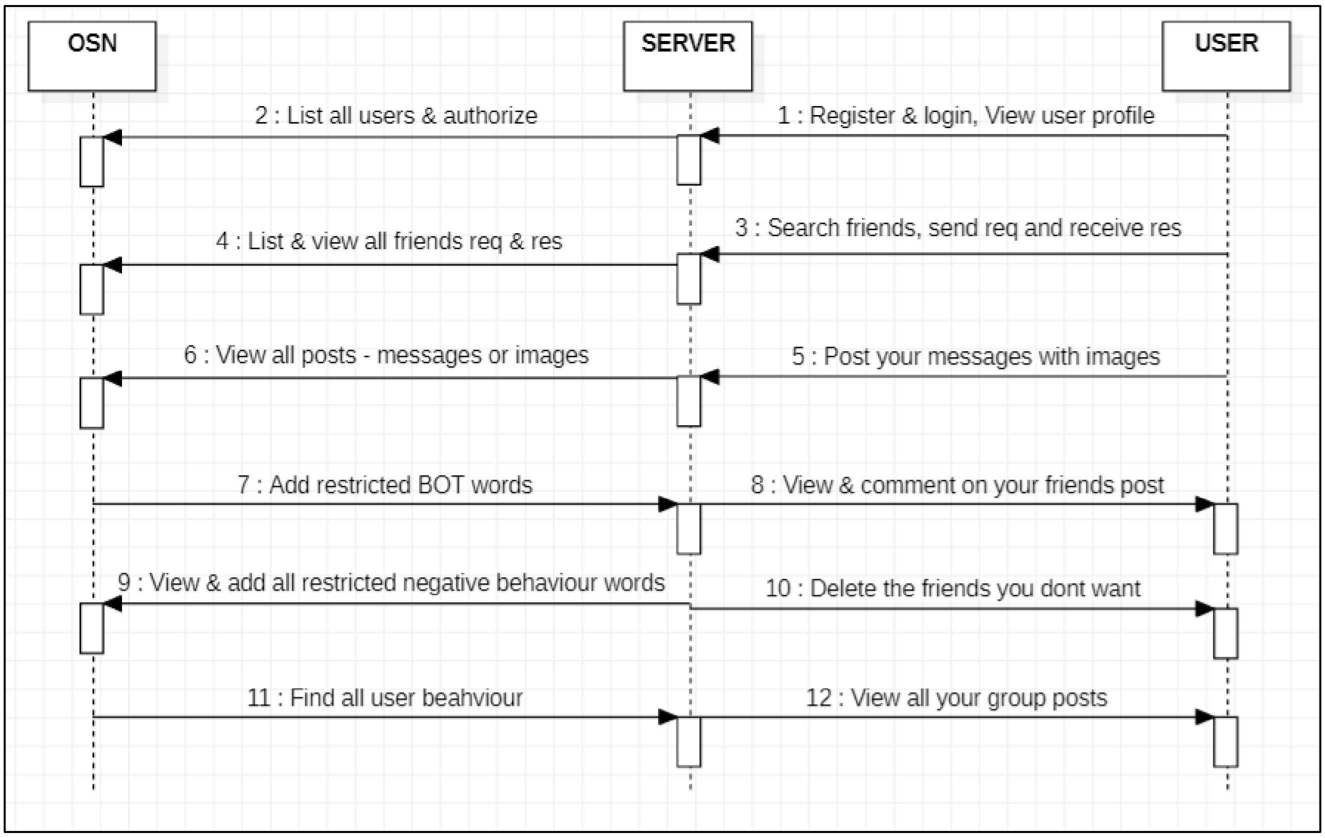

OSN
SERVER
USER
2 : List all users & authorize
1 : Register & login, View user profile
4 : List & view all friends req & res
3 : Search friends, send req and receive res
6 : View all posts - messages or images
5 : Post your messages with images
7 : Add restricted BOT words
8 : View & comment on your friends post
9 : View & add all restricted negative behaviour words
10 : Delete the friends you dont want
11 : Find all user beahviour
12 : View all your group posts

Fig 4.3.3: Sequence Diagram for Detecting Malicious Social Bots using Clickstream Sequences

4.3.4 Activity Diagram for Detecting Malicious Social Bots using Clickstream Sequences:

An Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

i. Administrator:

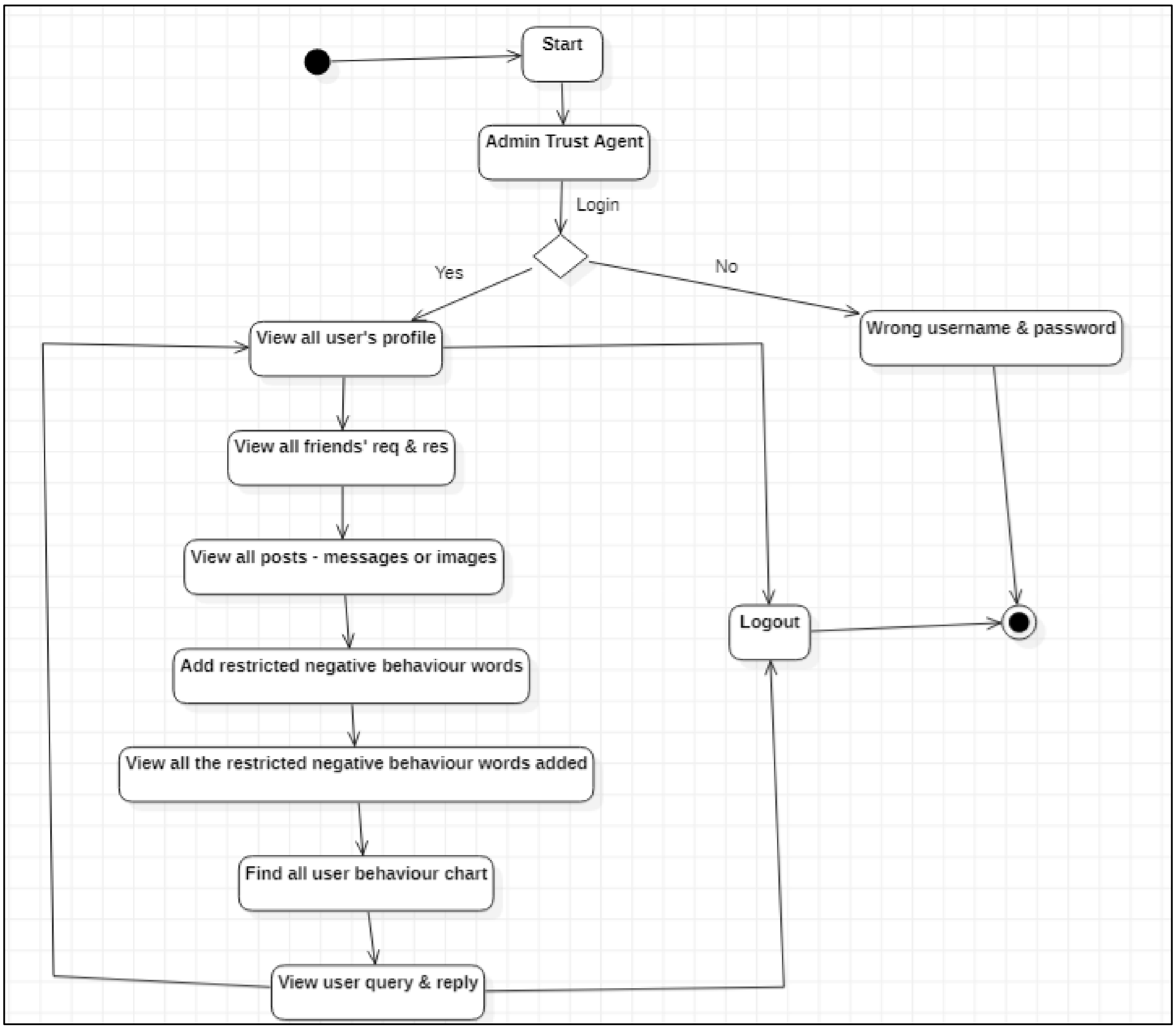

Start
Admin Trust Agent
Login
Yes
No
View all user's profile
Wrong username & password
View all friends' req & res
View all posts - messages or images
Logout
Add restricted negative behaviour words
View all the restricted negative behaviour words added
Find all user behaviour chart
View user query & reply

Fig 4.3.4.1: Activity Diagram (Administrator) for Detecting Malicious Social Bots using Clickstream Sequences

ii. User:

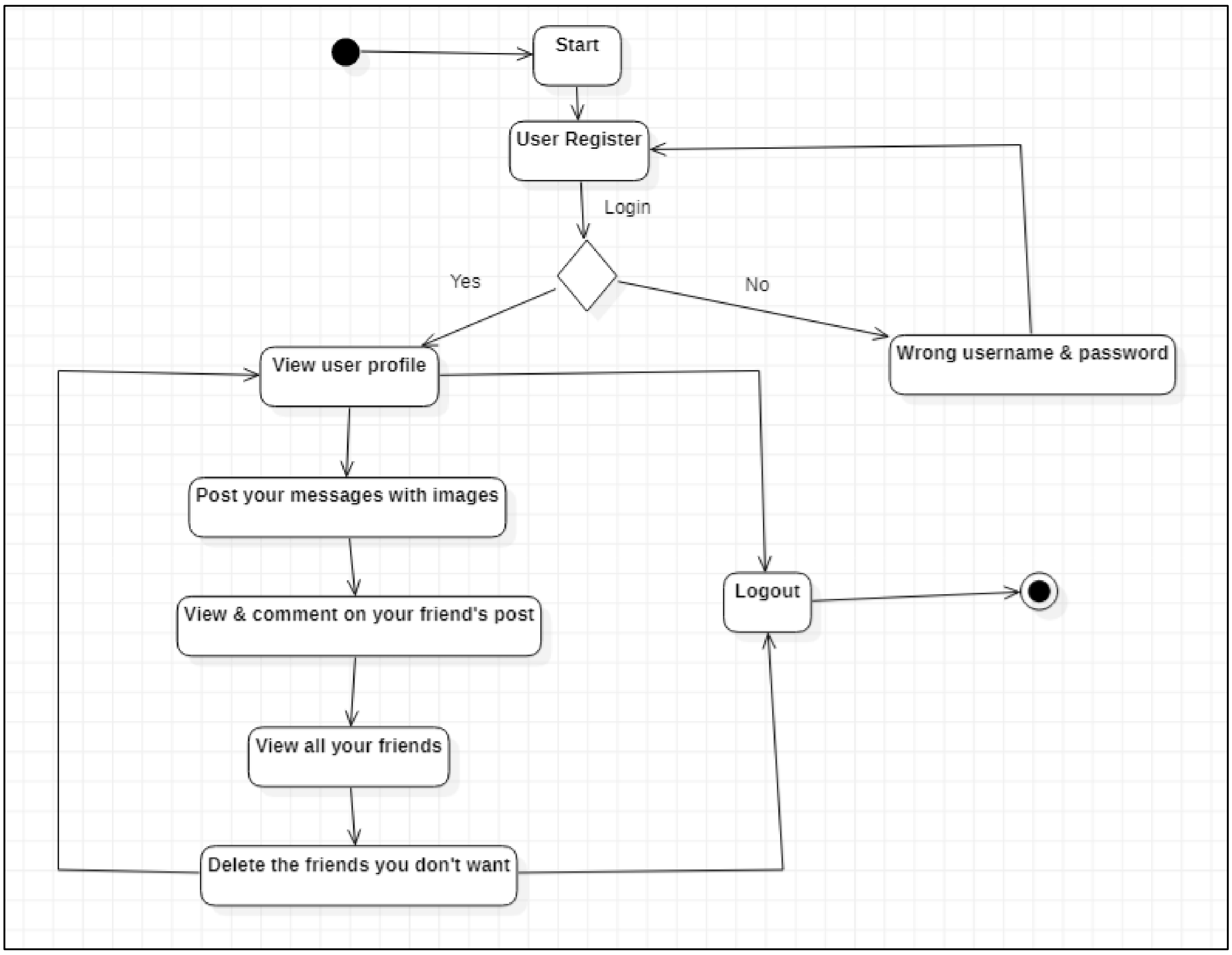

Start
User Register
Login
Yes
No
Wrong username & password
View user profile
Post your messages with images
View & comment on your friend's post
View all your friends
Delete the friends you don't want
Logout

Fig 4.3.4.2: Activity Diagram (User) for Detecting Malicious Social Bots using Clickstream Sequences

4.4 DATABASE DESIGN & TABLE DETAILS

Database design is the organization of data according to the database model. The designer determines what data must be stored and how the data elements interrelate. With this information we can fit the data into the database. It involves classifying data and identifying the interrelationships. It basically involves tables: lists of rows and columns. Each row is more correctly called a record and each column a field. A record is a meaningful and consistent way to combine information about something. A field is a single term of information- an item type that appears in every record. A key is a set of columns that is used to uniquely identify a record in a table. It is used to fetch or retrieve records/ data-rows from data table.

Certain principles guide the database design process. The design process consists of the following steps:

- Determining the purpose of the database
- Find and organise the information required
- Divide the information into tables
- Turn information items into columns
- Specify primary keys
- Set up the table relationships
- Refine your design
- Apply the normalization rules

A good database design is therefore the one that,

- Divides the information into subject-based tables to reduce redundant data.
- Provides access with the information it requires to join the information in the tables together as needed.
- Helps support and ensure the accuracy and integrity of the information.
- Accommodates the data processing and reporting needs.

Viewing the Database:

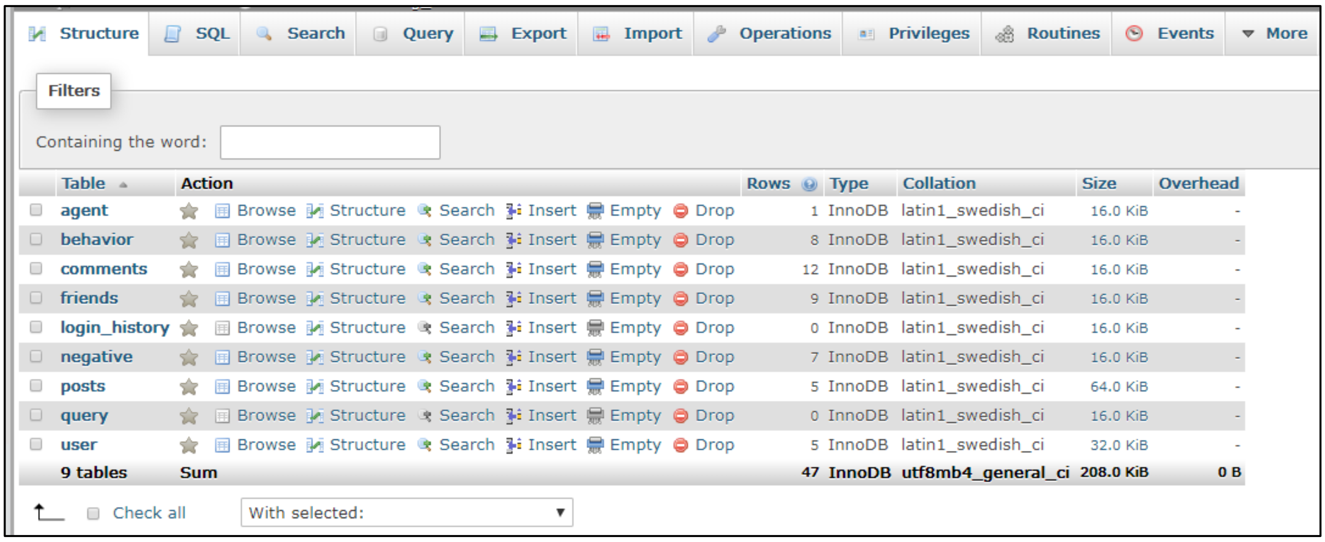
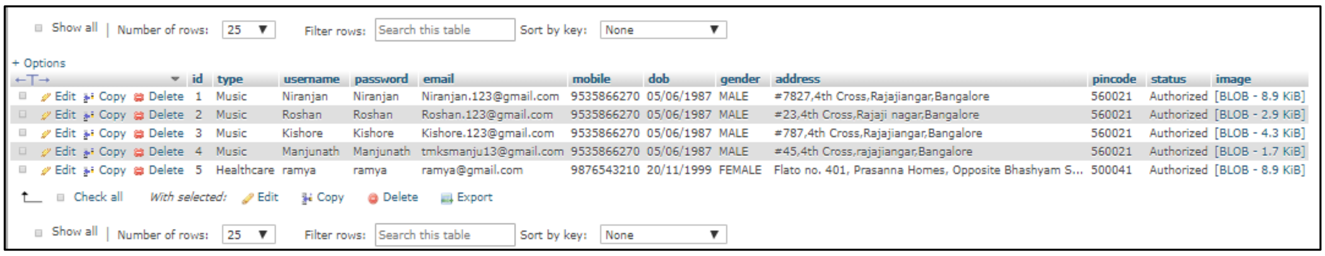


Table	Action	Rows	Type	Collation	Size	Overhead
agent	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
behavior	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	16.0 KiB	-
comments	Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16.0 KiB	-
friends	Browse Structure Search Insert Empty Drop	9	InnoDB	latin1_swedish_ci	16.0 KiB	-
login_history	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
negative	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16.0 KiB	-
posts	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	64.0 KiB	-
query	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
user	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	32.0 KiB	-
9 tables	Sum	47	InnoDB	utf8mb4_general_ci	208.0 KiB	0 B

Check all With selected:

Fig 4.4.1: database

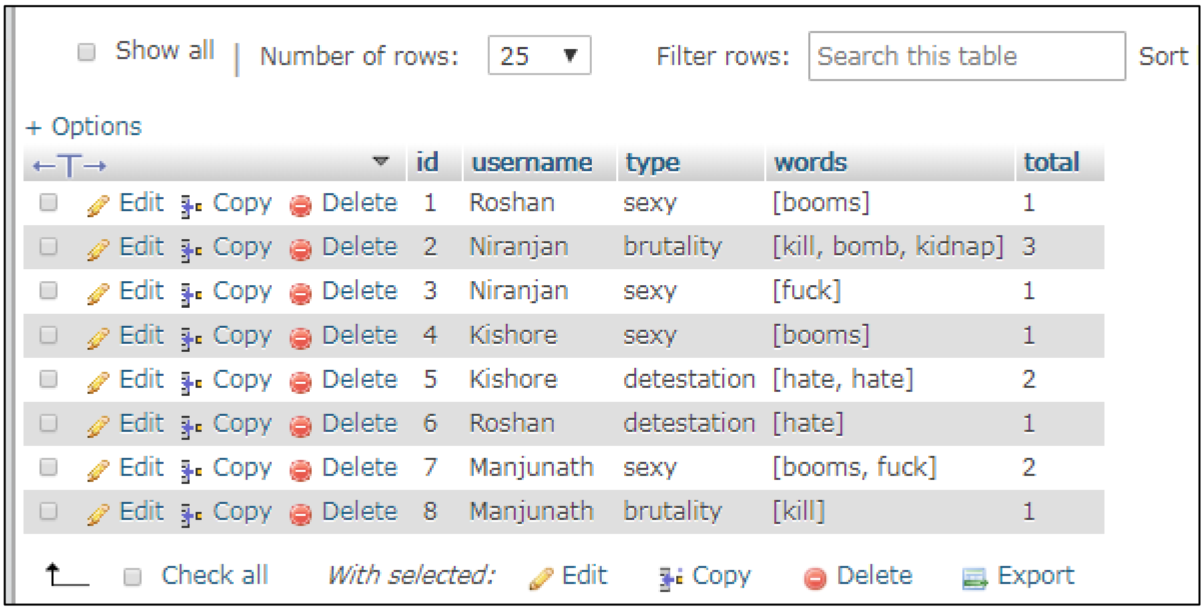
Viewing the "User" table:


Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected: Edit Copy Delete Export

	id	type	username	password	email	mobile	dob	gender	address	pincode	status	image
Edit Copy Delete	1	Music	Niranjan	Niranjan	Niranjan.123@gmail.com	9535866270	05/06/1987	MALE	#7827,4th Cross,Rajajiangar,Bangalore	560021	Authorized	[BLOB - 8.9 KiB]
Edit Copy Delete	2	Music	Roshan	Roshan	Roshan.123@gmail.com	9535866270	05/06/1987	MALE	#23,4th Cross,Rajaji nagar,Bangalore	560021	Authorized	[BLOB - 2.9 KiB]
Edit Copy Delete	3	Music	Kishore	Kishore	Kishore.123@gmail.com	9535866270	05/06/1987	MALE	#787,4th Cross,Rajajiangar,Bangalore	560021	Authorized	[BLOB - 4.3 KiB]
Edit Copy Delete	4	Music	Manjunath	Manjunath	tmksmanju13@gmail.com	9535866270	05/06/1987	MALE	#45,4th Cross,rajajiangar,Bangalore	560021	Authorized	[BLOB - 1.7 KiB]
Edit Copy Delete	5	Healthcare	ramya	ramya	ramya@gmail.com	9876543210	20/11/1999	FEMALE	Flato no. 401, Prasanna Homes, Opposite Bhashyam S...	500041	Authorized	[BLOB - 8.9 KiB]

Fig 4.4.2: user table

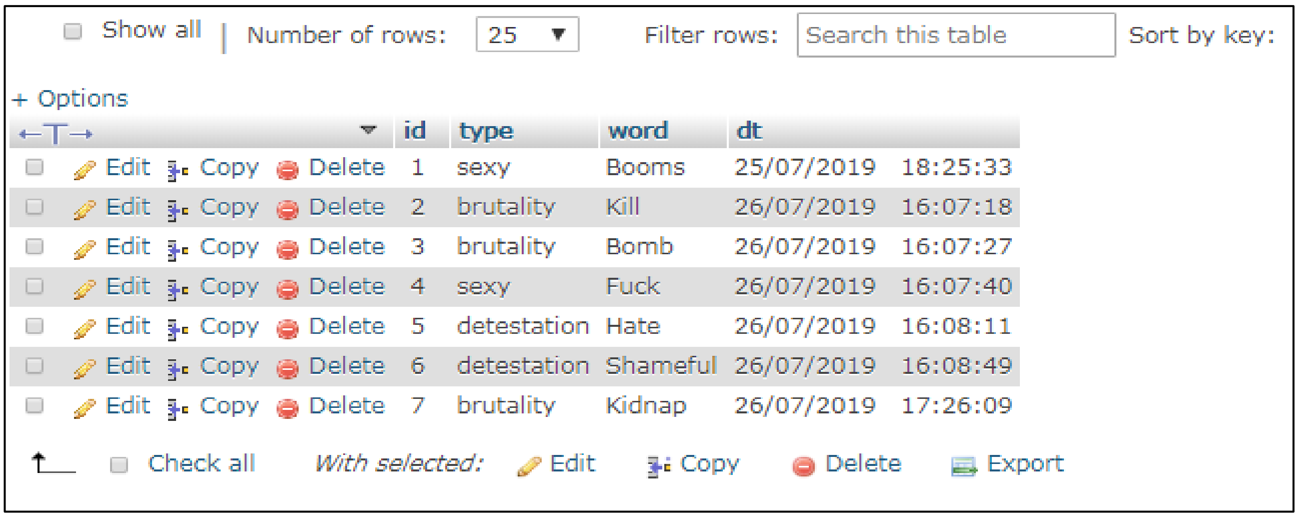
Viewing the "User Behaviour" table:


Show all
Number of rows: 25
Filter rows: Search this table
Sort
+ Options
Check all
With selected: Edit Copy Delete Export

	id	username	type	words	total
Edit Copy Delete	1	Roshan	sexy	[booms]	1
Edit Copy Delete	2	Niranjan	brutality	[kill, bomb, kidnap]	3
Edit Copy Delete	3	Niranjan	sexy	[fuck]	1
Edit Copy Delete	4	Kishore	sexy	[booms]	1
Edit Copy Delete	5	Kishore	detestation	[hate, hate]	2
Edit Copy Delete	6	Roshan	detestation	[hate]	1
Edit Copy Delete	7	Manjunath	sexy	[booms, fuck]	2
Edit Copy Delete	8	Manjunath	brutality	[kill]	1

Fig 4.4.3: user behaviour table

Viewing the “BOT words” table:



Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key:

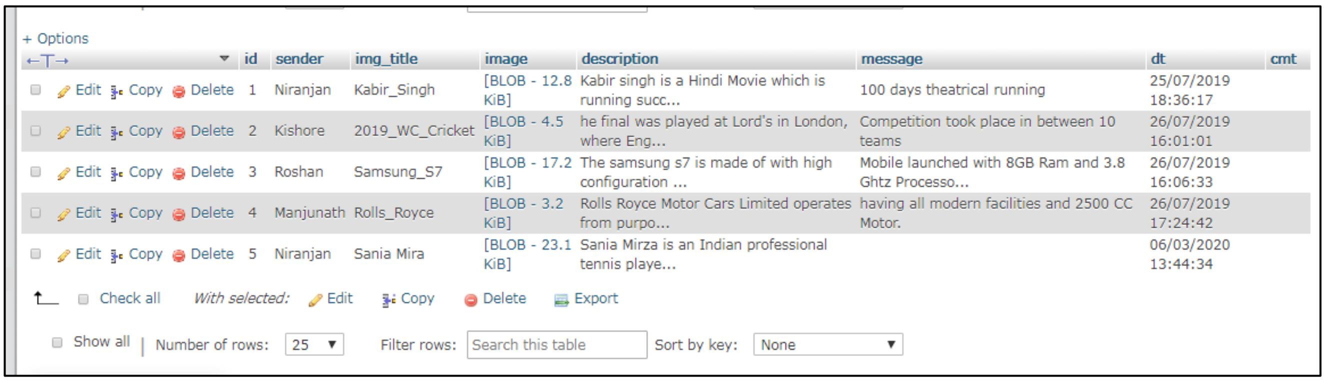
+ Options

←T→	id	type	word	dt
Edit Copy Delete	1	sexy	Booms	25/07/2019 18:25:33
Edit Copy Delete	2	brutality	Kill	26/07/2019 16:07:18
Edit Copy Delete	3	brutality	Bomb	26/07/2019 16:07:27
Edit Copy Delete	4	sexy	Fuck	26/07/2019 16:07:40
Edit Copy Delete	5	detestation	Hate	26/07/2019 16:08:11
Edit Copy Delete	6	detestation	Shameful	26/07/2019 16:08:49
Edit Copy Delete	7	brutality	Kidnap	26/07/2019 17:26:09

Check all *With selected:* Edit Copy Delete Export

Fig 4.4.4: BOT words table

Viewing “All posts” table:



+ Options

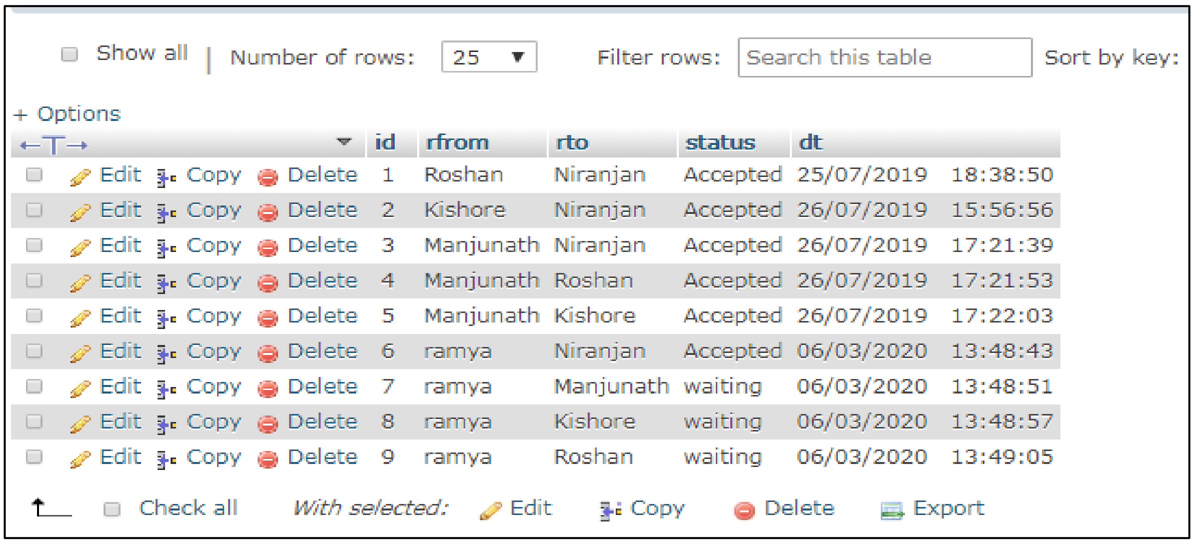
←T→	id	sender	img_title	image	description	message	dt	cmt
Edit Copy Delete	1	Niranjan	Kabir_Singh	[BLOB - 12.8 KiB]	Kabir singh is a Hindi Movie which is running succ...	100 days theatrical running	25/07/2019 18:36:17	
Edit Copy Delete	2	Kishore	2019_WC_Cricket	[BLOB - 4.5 KiB]	he final was played at Lord's in London, where Eng...	Competition took place in between 10 teams	26/07/2019 16:01:01	
Edit Copy Delete	3	Roshan	Samsung_S7	[BLOB - 17.2 KiB]	The samsung s7 is made of with high configuration ...	Mobile launched with 8GB Ram and 3.8 Ghtz Processo...	26/07/2019 16:06:33	
Edit Copy Delete	4	Manjunath	Rolls_Royce	[BLOB - 3.2 KiB]	Rolls Royce Motor Cars Limited operates from purpo...	having all modern facilities and 2500 CC Motor.	26/07/2019 17:24:42	
Edit Copy Delete	5	Niranjan	Sania Mira	[BLOB - 23.1 KiB]	Sania Mirza is an Indian professional tennis playe...		06/03/2020 13:44:34	

Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key: None ▼

Fig 4.4.5: all posts table

Viewing the “Friend Requests” table:



Show all | Number of rows: 25 ▼ Filter rows: Search this table Sort by key:

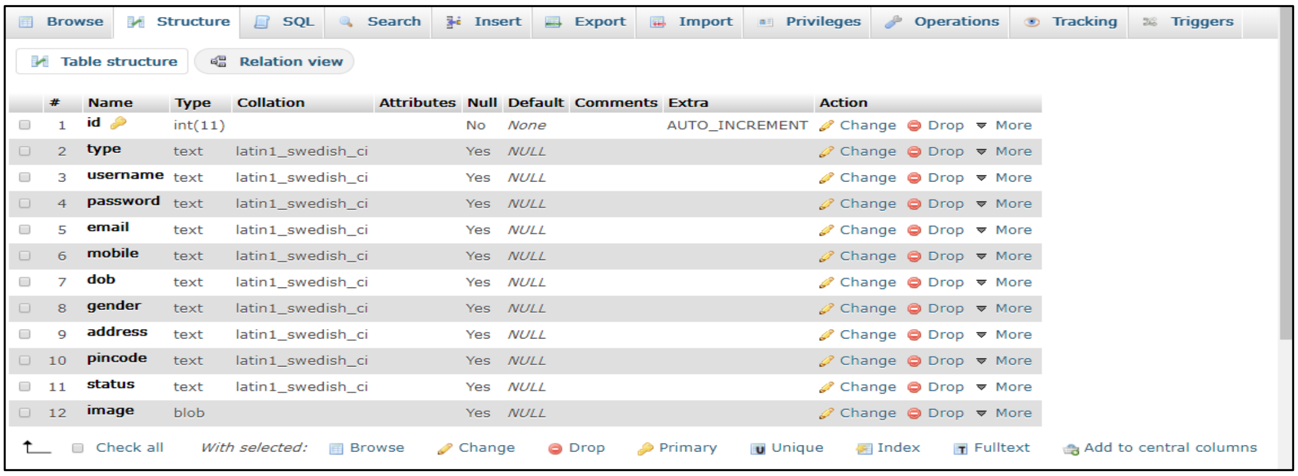
+ Options

←T→	id	rfrom	rto	status	dt
Edit Copy Delete	1	Roshan	Niranjan	Accepted	25/07/2019 18:38:50
Edit Copy Delete	2	Kishore	Niranjan	Accepted	26/07/2019 15:56:56
Edit Copy Delete	3	Manjunath	Niranjan	Accepted	26/07/2019 17:21:39
Edit Copy Delete	4	Manjunath	Roshan	Accepted	26/07/2019 17:21:53
Edit Copy Delete	5	Manjunath	Kishore	Accepted	26/07/2019 17:22:03
Edit Copy Delete	6	ramya	Niranjan	Accepted	06/03/2020 13:48:43
Edit Copy Delete	7	ramya	Manjunath	waiting	06/03/2020 13:48:51
Edit Copy Delete	8	ramya	Kishore	waiting	06/03/2020 13:48:57
Edit Copy Delete	9	ramya	Roshan	waiting	06/03/2020 13:49:05

Check all *With selected:* Edit Copy Delete Export

Fig 4.4.6: friend requests table

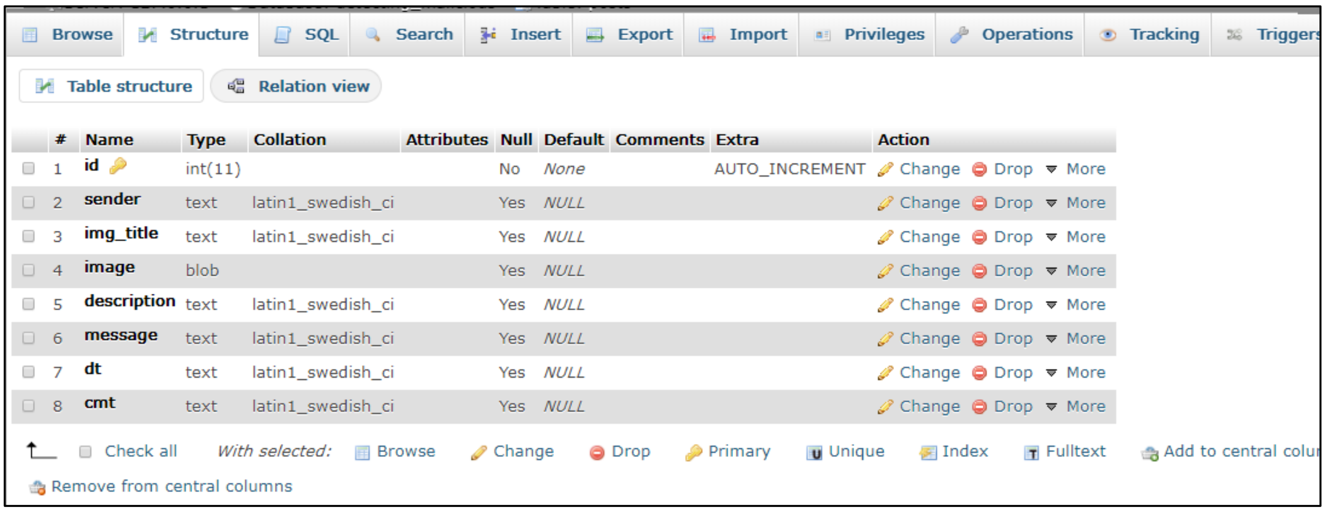
Structure of “user” table:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	type	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
3	username	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
4	password	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
5	email	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
6	mobile	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
7	dob	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
8	gender	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
9	address	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
10	pincode	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
11	status	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
12	image	blob			Yes	*NULL*			Change Drop More

Fig 4.4.7: structure of user table

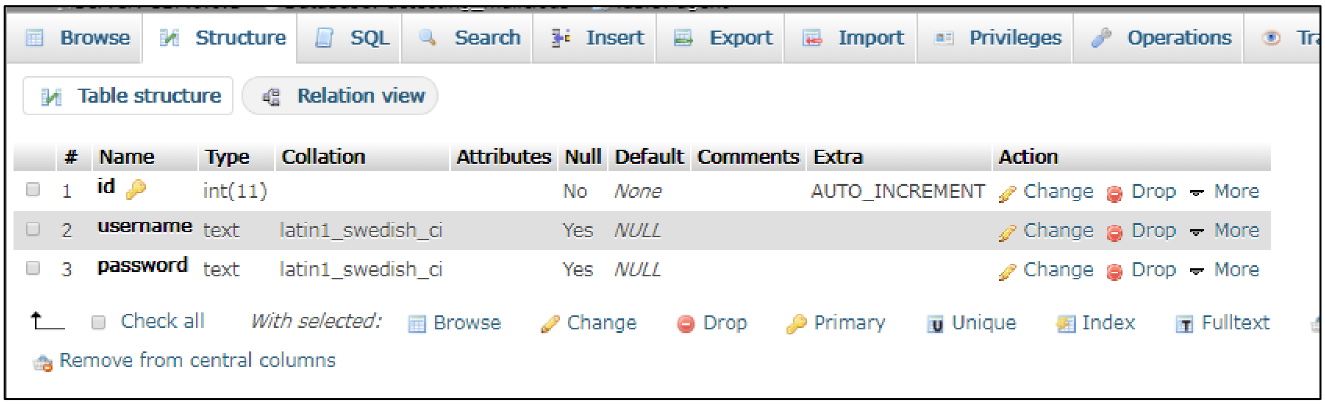
Structure of “Posts” table:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	sender	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
3	img_title	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
4	image	blob			Yes	*NULL*			Change Drop More
5	description	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
6	message	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
7	dt	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
8	cmt	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More

Fig 4.4.8: structure of posts table

Structure of “Admin” table:



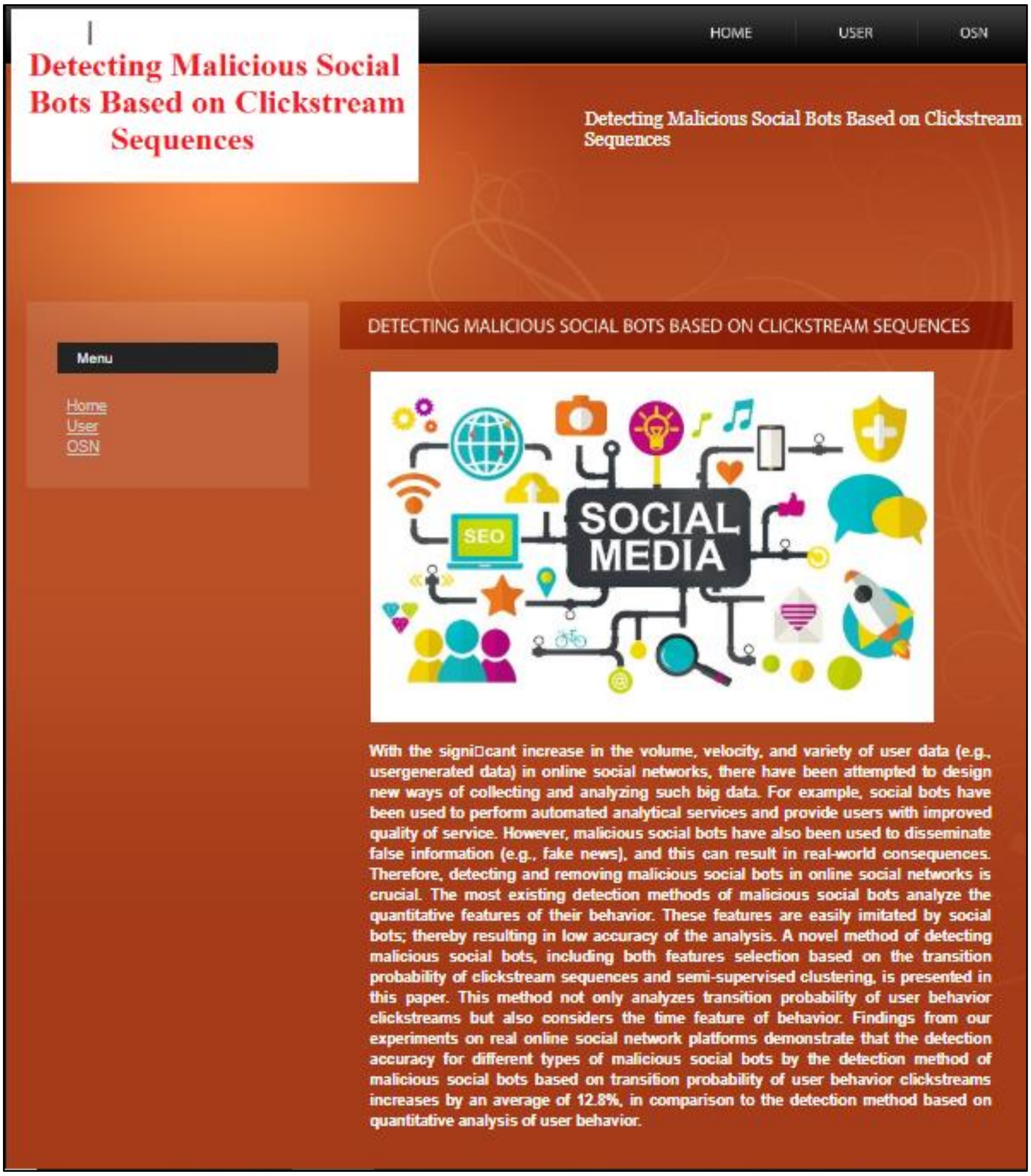
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
2	username	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
3	password	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More

Fig 4.4.9: structure of admin table

5. USER INTERFACE & CODE IMPLEMENTATION

5.1 WEBPAGES DETAILS

Home Page: This is the home page, where the user or the OSN Admin can login and perform their activities.


Detecting Malicious Social Bots Based on Clickstream Sequences
HOME
USER
OSN
Menu
Home
User
OSN
SOCIAL MEDIA
SEO

Detecting Malicious Social Bots Based on Clickstream Sequences

DETECTING MALICIOUS SOCIAL BOTS BASED ON CLICKSTREAM SEQUENCES

With the signi□cant increase in the volume, velocity, and variety of user data (e.g., usergenerated data) in online social networks, there have been attempted to design new ways of collecting and analyzing such big data. For example, social bots have been used to perform automated analytical services and provide users with improved quality of service. However, malicious social bots have also been used to disseminate false information (e.g., fake news), and this can result in real-world consequences. Therefore, detecting and removing malicious social bots in online social networks is crucial. The most existing detection methods of malicious social bots analyze the quantitative features of their behavior. These features are easily imitated by social bots; thereby resulting in low accuracy of the analysis. A novel method of detecting malicious social bots, including both features selection based on the transition probability of clickstream sequences and semi-supervised clustering, is presented in this paper. This method not only analyzes transition probability of user behavior clickstreams but also considers the time feature of behavior. Findings from our experiments on real online social network platforms demonstrate that the detection accuracy for different types of malicious social bots by the detection method of malicious social bots based on transition probability of user behavior clickstreams increases by an average of 12.8%, in comparison to the detection method based on quantitative analysis of user behavior.

Fig 5.1.1: home page

User Login: The user can login by going to this page. If a user does not have an account, he can REGISTER by clinking on "Register" button shown.

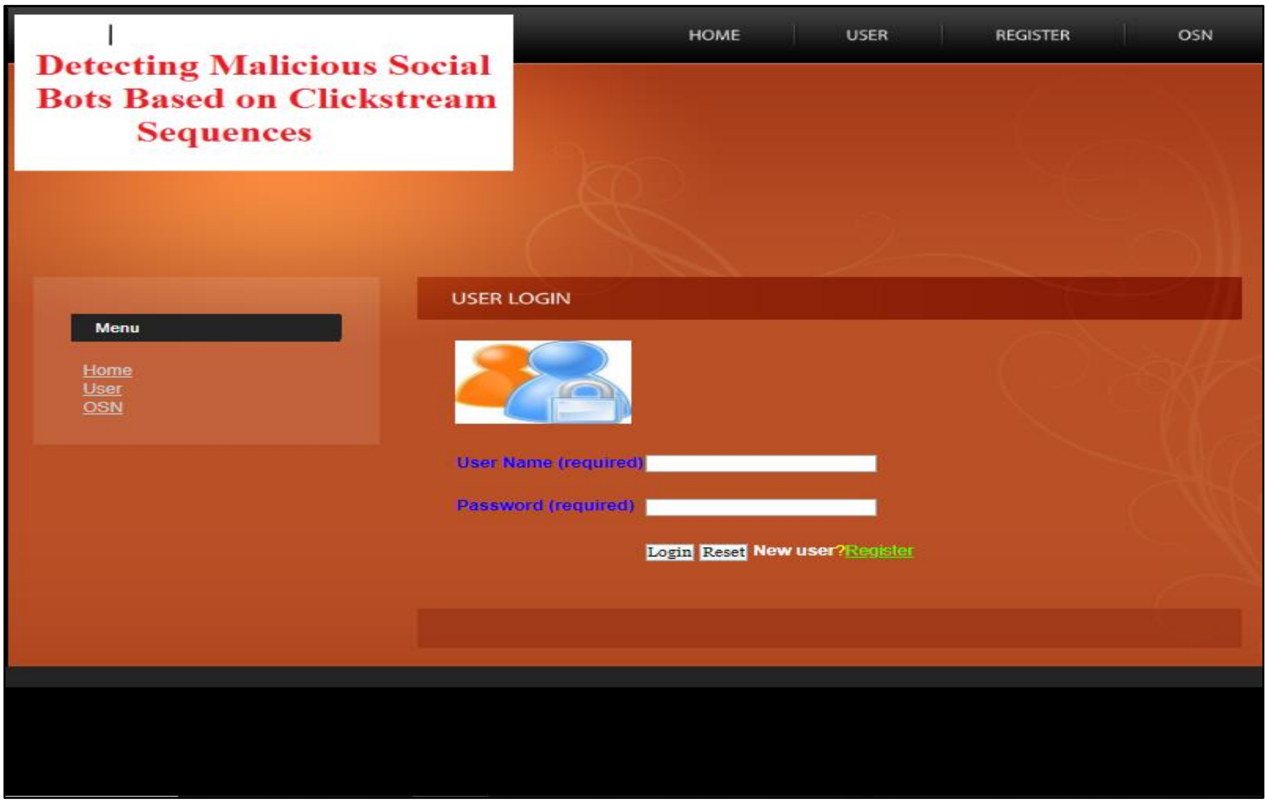


Fig 5.1.2: user login

User Register: If the user does not have an account, he can register clicking on "Register" and the following page views:

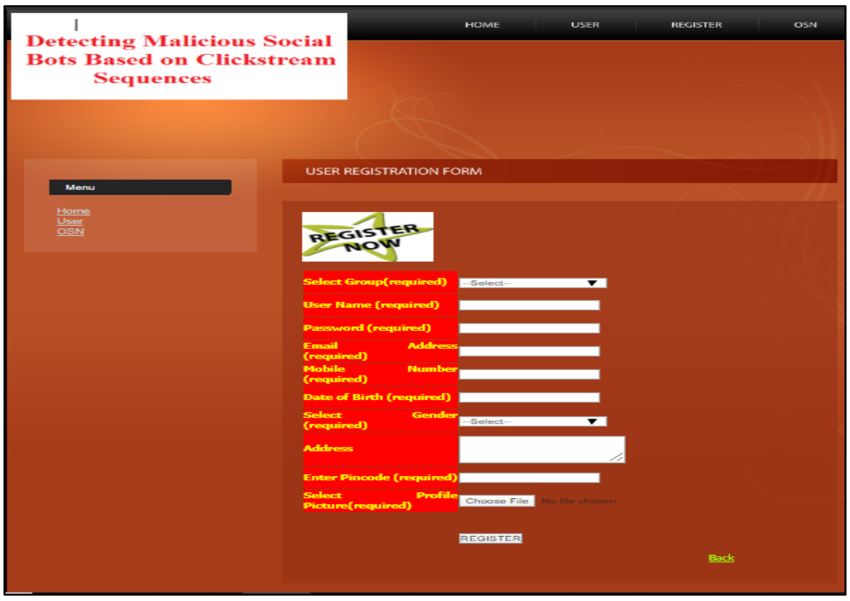


Fig 5.1.3: user register

User Home Page: This is the User main page. The user can view his profile, search his friends, and send them friend requests, accept friend requests from others, post any message, view and comment on his friend's posts.

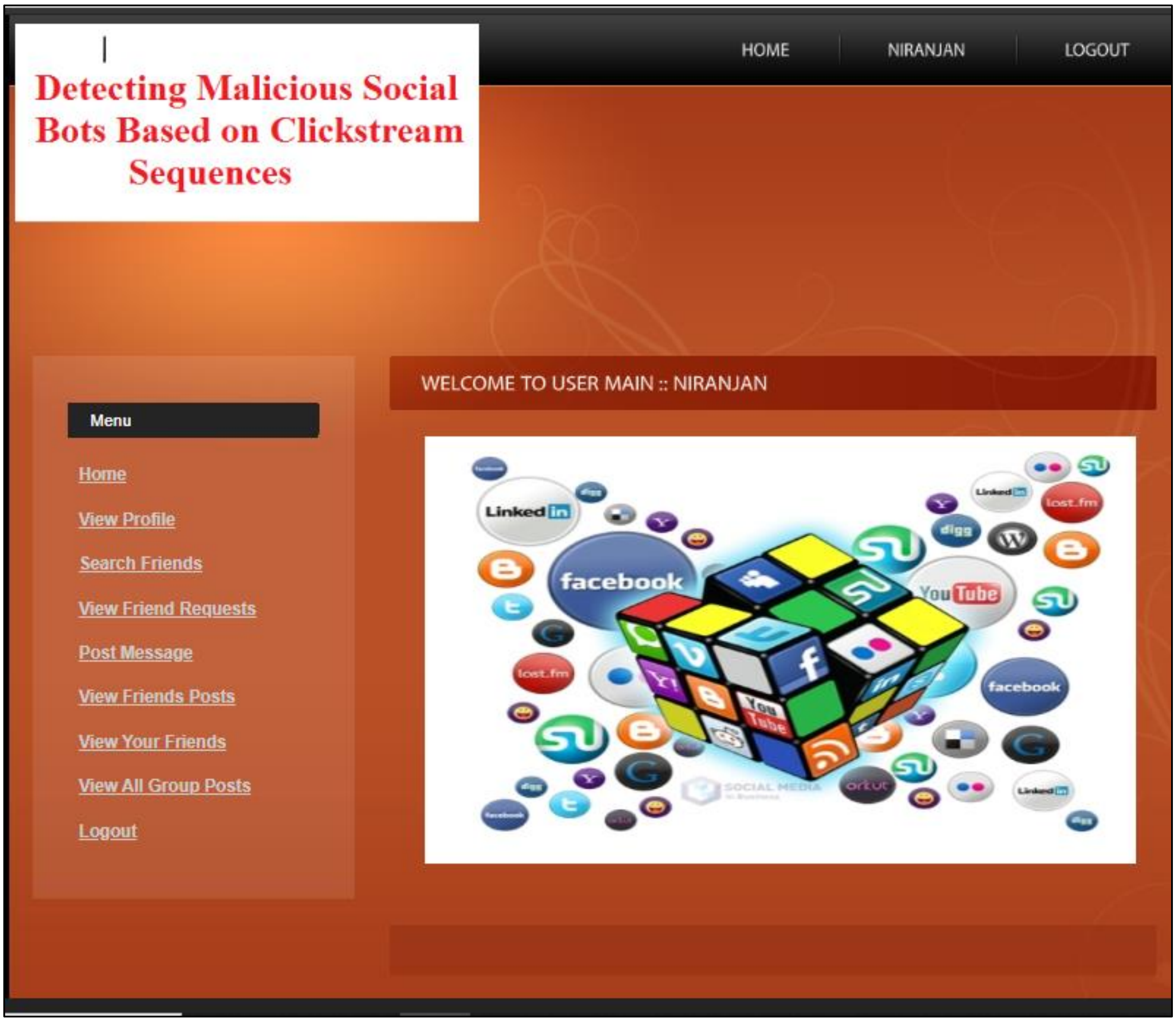
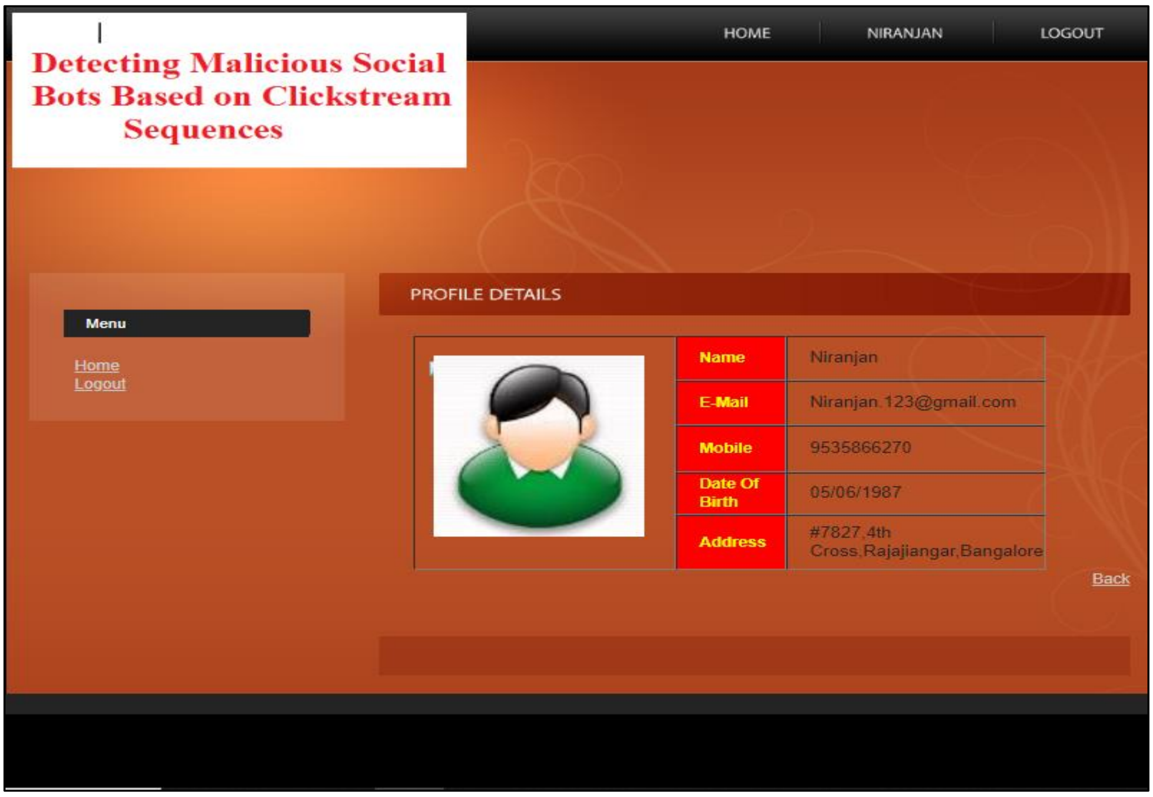


Fig 5.1.4: user home page

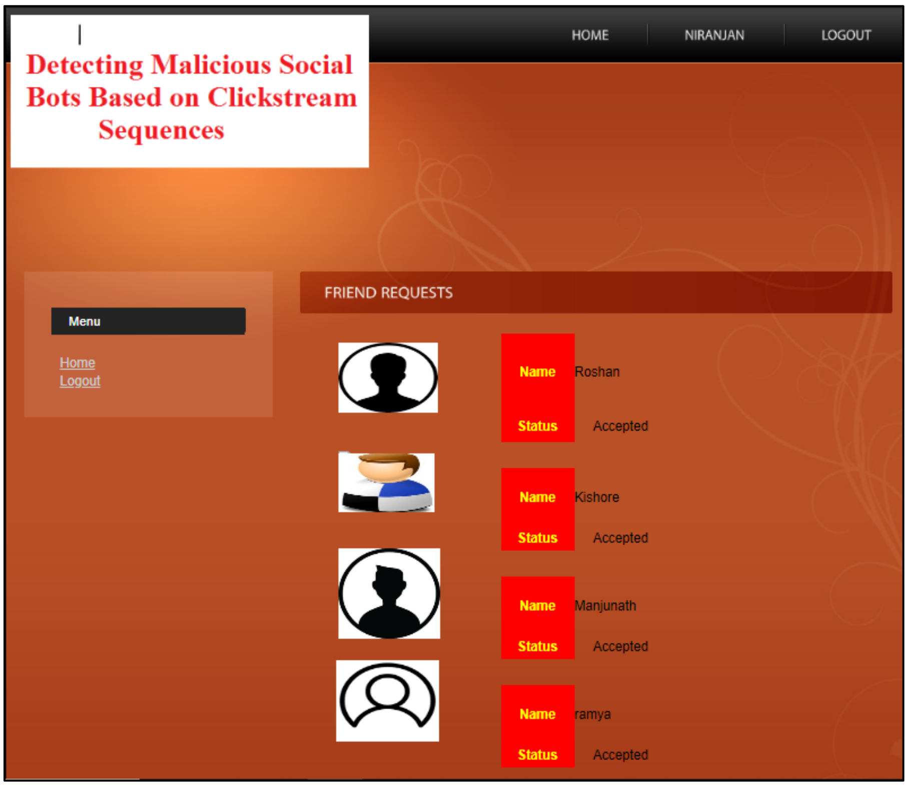
User Profile: The user can view his profile and can edit according to his choice.


Detecting Malicious Social Bots Based on Clickstream Sequences
HOME
NIRANJAN
LOGOUT
Menu
Home
Logout
PROFILE DETAILS
Back

Name	Niranjan
E-Mail	Niranjan.123@gmail.com
Mobile	9535866270
Date Of Birth	05/06/1987
Address	#7827,4th Cross,Rajajiangar,Bangalore

Fig 5.1.5: user profile

User Friends: The user can view all his friends and friend requests he has received.


Detecting Malicious Social Bots Based on Clickstream Sequences
HOME
NIRANJAN
LOGOUT
Menu
Home
Logout
FRIEND REQUESTS

Name Roshan
Status Accepted

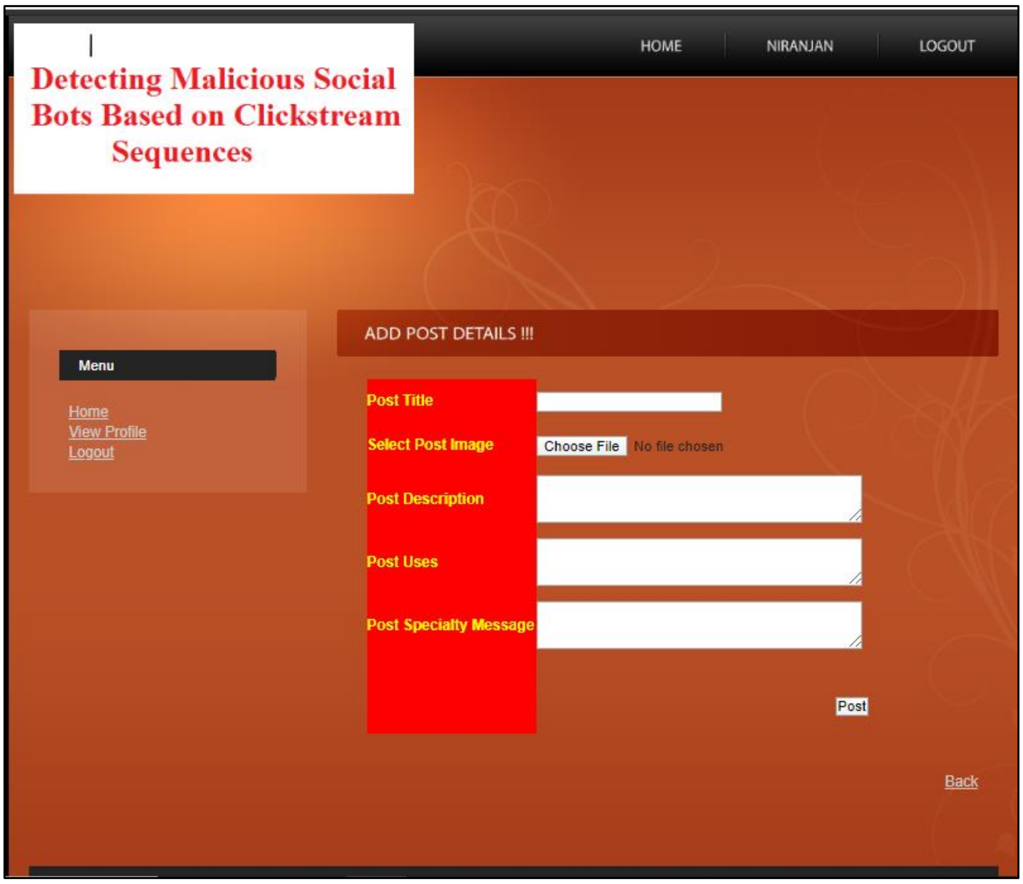
Name Kishore
Status Accepted

Name Manjunath
Status Accepted

Name ramya
Status Accepted

Fig 5.1.6: user friends

Add a New Post: The user can add a new post with an image.

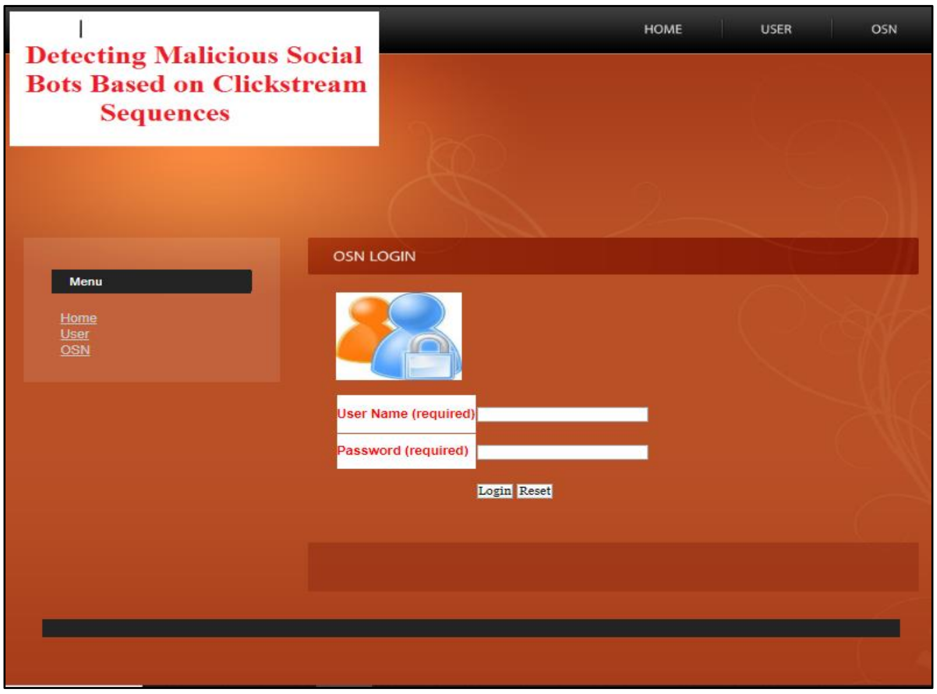

Detecting Malicious Social Bots Based on Clickstream Sequences

ADD POST DETAILS !!!

Post Title
Select Post Image Choose File No file chosen
Post Description
Post Uses
Post Specialty Message

Fig 5.1.7: add post

Admin Login: The admin can login by using this page, where he needs a username and password.


Detecting Malicious Social Bots Based on Clickstream Sequences

OSN LOGIN

User Name (required)
Password (required)

Fig 5.1.8: admin login

Admin Home Page: After the Admin logs into his account, the following is viewed. The admin can view all users, their posts, can add BOT words, detects those who use the BOT words, keeps track of the user behaviour on the server.

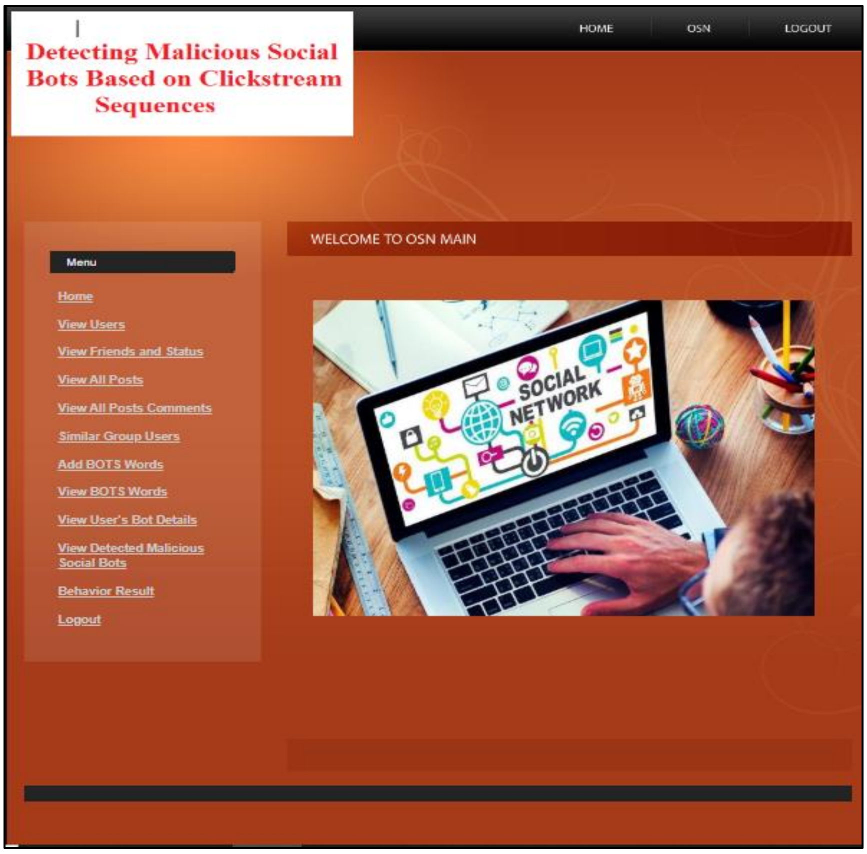


Fig 5.1.9: admin home page

User behaviour: The admin prepares a chart of the user behaviour, whether they are malicious or not. This chart helps us in analysing the users who are fake and who are real, to avoid spread of wrong information.

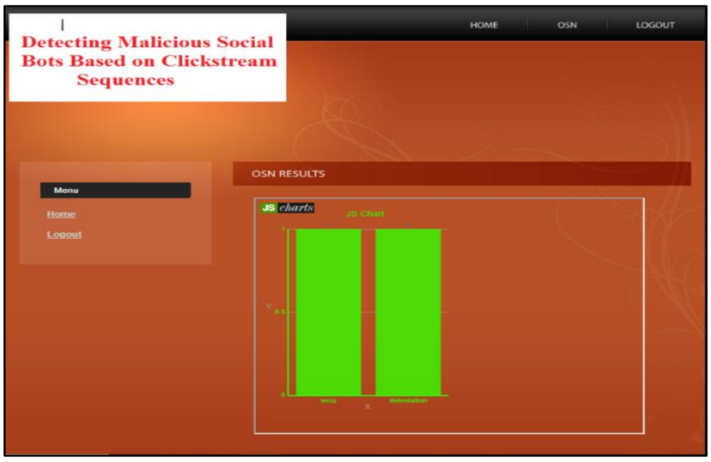


Fig 5.1.10: user behaviour

5.2 SAMPLE CODE

5.2.1 userlogin.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>users login page</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script src="js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script src="js/cufon-yui.js" type="text/javascript"></script>
<script src="js/cufon-replace.js" type="text/javascript"></script>
<script src="js/Myriad_Pro_400.font.js" type="text/javascript"></script>
<!--[if lt IE 7]>
<link href="ie_style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="js/ie_png.js"></script>
<script type="text/javascript">ie_png.fix('.png, .extra-bg, .box, .box1, .box2, .img-list img');</script>
<![endif]-->
<style type="text/css">
<!--
.style1 {
	font-size: 36px;
	font-family: Georgia, "Times New Roman", Times, serif;
	color: #FF3366;
}
.style2 {color: #999999}
.style3 {font-family: Georgia, "Times New Roman", Times, serif}
.style4 {color: #33FF00; }
.style6 {color: #0066FF}
.style7 {color: #FFFF00}
```

```
.style10 {color: #FFFFFF; font-weight: bold; }
.style15 {color: #0000FF; font-weight: bold; }
-->
</style>
</head>
<body id="page1">
<div id="main">
  <div class="extra-bg"></div>
  <!-- header -->
  <div id="header">
    <div class="row-1">
      <ul class="top-links">
        <li><a href="#"><img alt="" src="images/home-icon.gif" /></a></li>
        <li><a href="#"><img alt="" src="images/mail-icon.gif" /></a></li>
      </ul>
      <ul class="nav">
        <li class="first"><a href="index.html">Home</a></li>
              <li><a href="userlogin.jsp"><span>User</span></a> </li>
              <li><a href="register.jsp"><span>Register</span></a> </li>
                    <li><a href="agentlogin.jsp">OSN</a></li>
      </ul>
    </div>
    <div class="row-2 style6">
      <p  class="first  style1">&nbsp;</p>
      </div>
  </div>
  <!-- content -->
  <div id="content">
    <div class="wrapper">
      <div class="aside">
```

```
        <div class="inner_copy">More <a href="#">Website Templates</a> @ Templates.com!</div>
```

5.2.2 connect.jsp

```
<%@ page import="java.sql.*"%>
<%@ page import="java.util.*" %>
<%
        Connection connection = null;
        try {

          Class.forName("com.mysql.jdbc.Driver");
        connection = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/Detecting_Malicious","root","root");
        String sql="";

        }
        catch(Exception e)
        {
        System.out.println(e);
        }
%>
```

6. CODE DEPLOYMENT IN THE CLOUD

6.1 INTRODUCTION

6.1.1 Deployment Models

Private Cloud:

Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third party, and hosted either internally or externally. Undertaking a private cloud project requires significant engagement to virtualize the business environment and requires the organization to reevaluate decisions about existing resources. It can improve business, but every step in the project raises security issues that must be addressed to prevent serious vulnerabilities. Self-run data centers are generally capital intensive. They have a significant physical footprint, requiring allocations of space, hardware, and environmental controls. These assets have to be refreshed periodically, resulting in additional capital expenditures. They have attracted criticism because users "still have to buy, build, and manage them" and thus do not benefit from less hands-on management, essentially "[lacking] the economic model that makes cloud computing such an intriguing concept".

Public Cloud:

A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free.[Technically there may be little or no difference between public and private cloud architecture, however, security consideration may be substantially different for services (applications, storage, and other resources) that are made available by a service provider for a public audience and when communication is effected over a non-trusted network. Generally, public cloud service providers like Amazon Web Services (AWS), IBM, Oracle, Microsoft, Google, and Alibaba own and operate the infrastructure at their data center and access is generally via the Internet. AWS, Oracle, Microsoft, and Google also offer direct connect services called "AWS Direct Connect", "Oracle FastConnect", "Azure ExpressRoute", and "Cloud Interconnect" respectively, such connections require customers to purchase or lease a private connection to a peering point offered by the cloud provider.

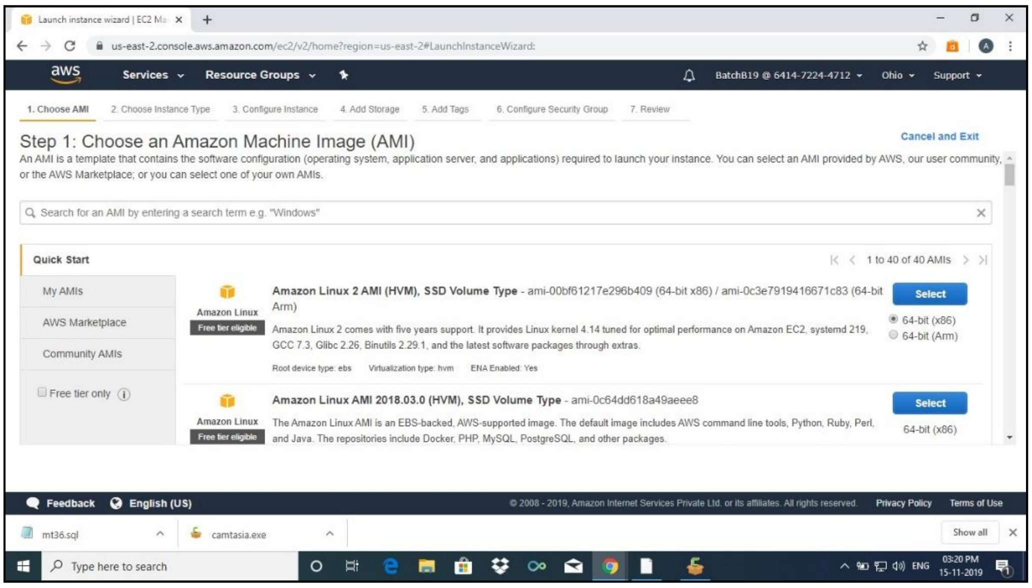
Hybrid Cloud:

Hybrid cloud is a composition of a public cloud and a private environment, such as a private cloud or on-premise resources, that remain distinct entities but are bound together, offering the benefits of multiple deployment models. Hybrid cloud can also mean the ability to connect collocation, managed and/or dedicated services with cloud resources. Gartner defines a hybrid cloud service as a cloud computing service that is composed of some combination of private, public and community cloud services, from different service providers. A hybrid cloud service crosses isolation and provider boundaries so that it cannot be simply put in one category of private, public, or community cloud service. It allows one to extend either the capacity or the capability of a cloud service, by aggregation, integration, or customization with another cloud service.

6.2 PROCEDURE TO DEPLOY

6.2.1 Creating an EC2 Instance in AWS

Step 1: First, choose an Amazon Machine Image (AMI). Here, we have selected Amazon Linux 2 AMI



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-00bf61217e296b409 (64-bit x86) / ami-0c3e7919416671c83 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

[✓ 64-bit (x86)] 64-bit (Arm)

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0c64dd618a49aeee8

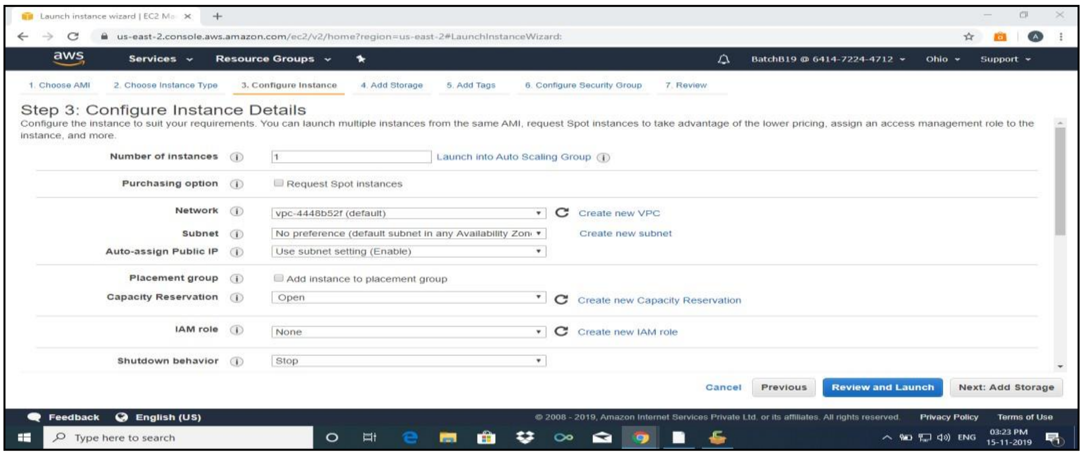
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

64-bit (x86)

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Fig 6.2.1.1: AMI

Step 2: The, choose the Instance type and config the Instance



Step 3: Configure Instance Details

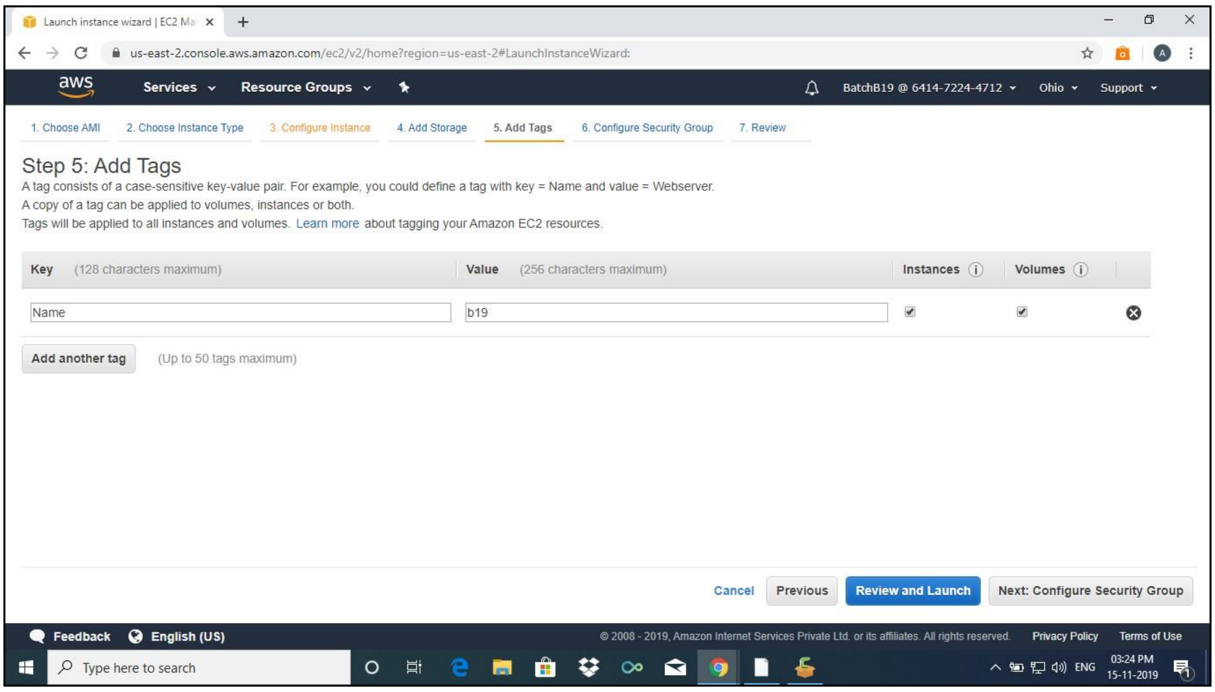
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	1
Purchasing option	Request Spot instances
Network	vpc-4448b52f (default)
Subnet	No preference (default subnet in any Availability Zon
Auto-assign Public IP	Use subnet setting (Enable)
Placement group	Add instance to placement group
Capacity Reservation	Open
IAM role	None
Shutdown behavior	Stop

Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Services
Resource Groups
BatchB19 @ 6414-7224-4712
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Launch into Auto Scaling Group
Create new VPC
Create new subnet
Create new Capacity Reservation
Create new IAM role
Cancel
Previous
Review and Launch
Next: Add Storage
Feedback
English (US)
© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
03:23 PM
15-11-2019

Fig 6.2.1.2: Configure the Instance

Step 3: Add Tags i.e., Add your Instance name



Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

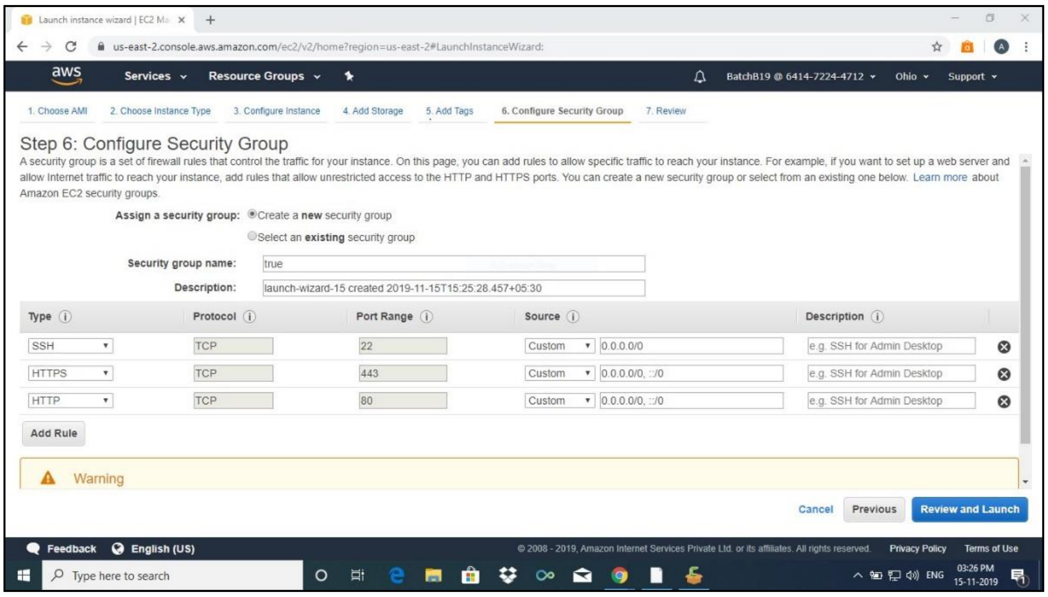
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	b19	✓	✓

Launch instance wizard | EC2 Ma
us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LaunchInstanceWizard:
Services
Resource Groups
BatchB19 @ 6414-7224-4712
Ohio
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Add another tag
(Up to 50 tags maximum)
Cancel
Previous
Review and Launch
Next: Configure Security Group
Feedback
English (US)
© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Type here to search
03:24 PM
15-11-2019

Fig 6.2.1.3: Add Tags

Step 4: Config the Security Group



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: [✓ Create a **new** security group] Select an **existing** security group

Security group name: true

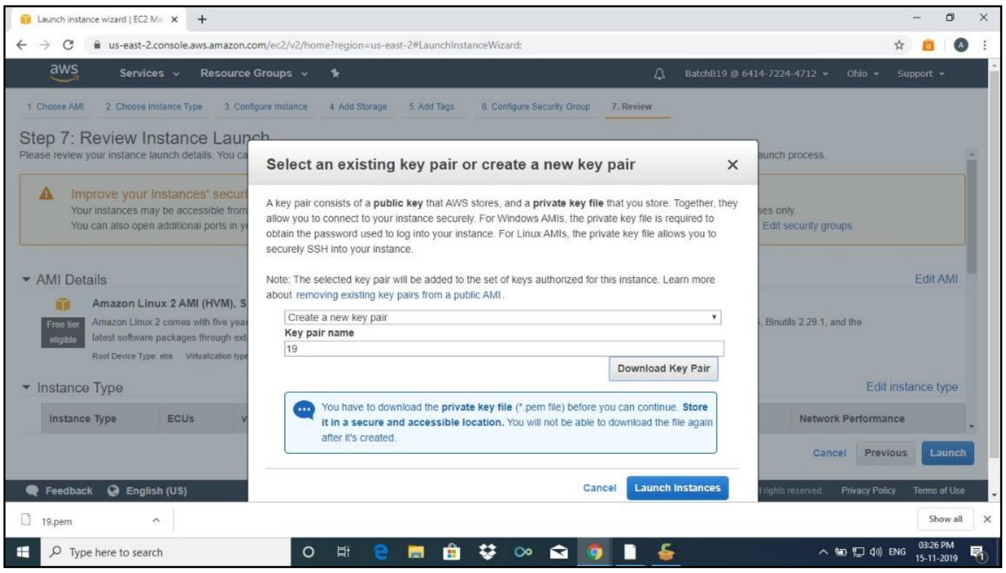
Description: launch-wizard-15 created 2019-11-15T15:25:28.457+05:30

Type	Protocol	Port Range	Source		Description
SSH	TCP	22	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Warning

Fig 6.2.1.4: Configure Security Group

Step 5: Now, Download the Key-Pair created



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

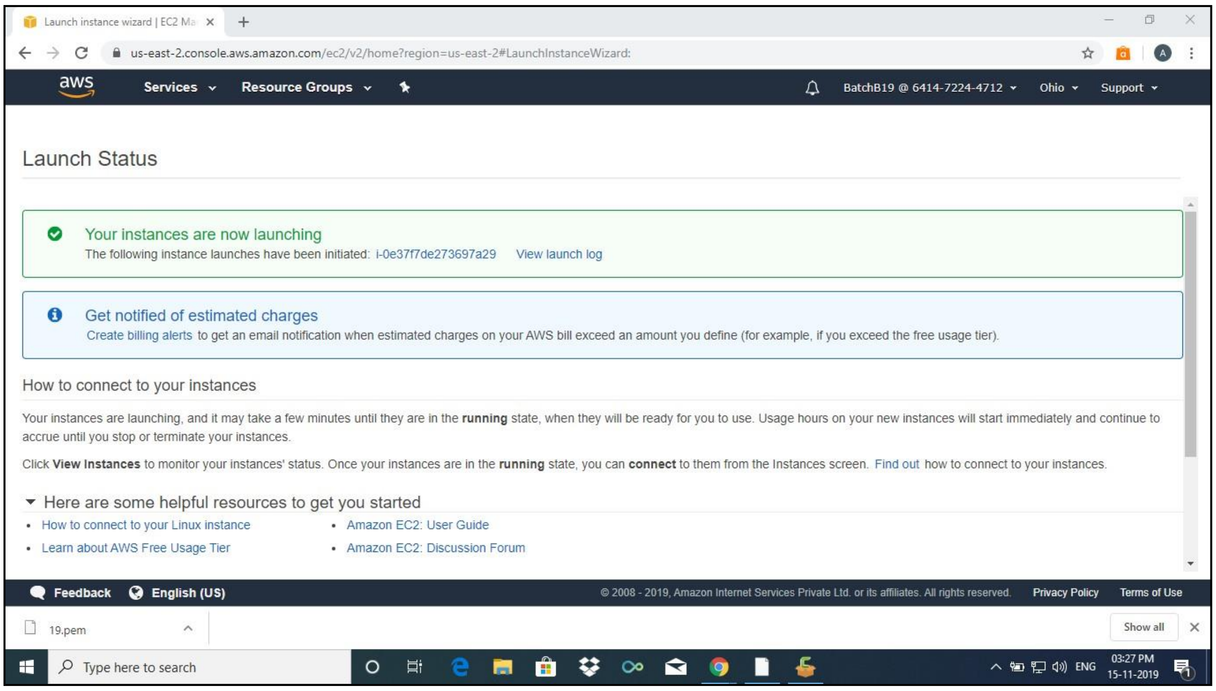
Create a new key pair

Key pair name: 19

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Fig 6.2.1.5: Download Key-Pair

Step 6: The, Review the details and Launch the Instance


Launch Status

Your instances are now launching
The following instance launches have been initiated: i-0e37f7de273697a29 View launch log

Get notified of estimated charges
Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances

Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

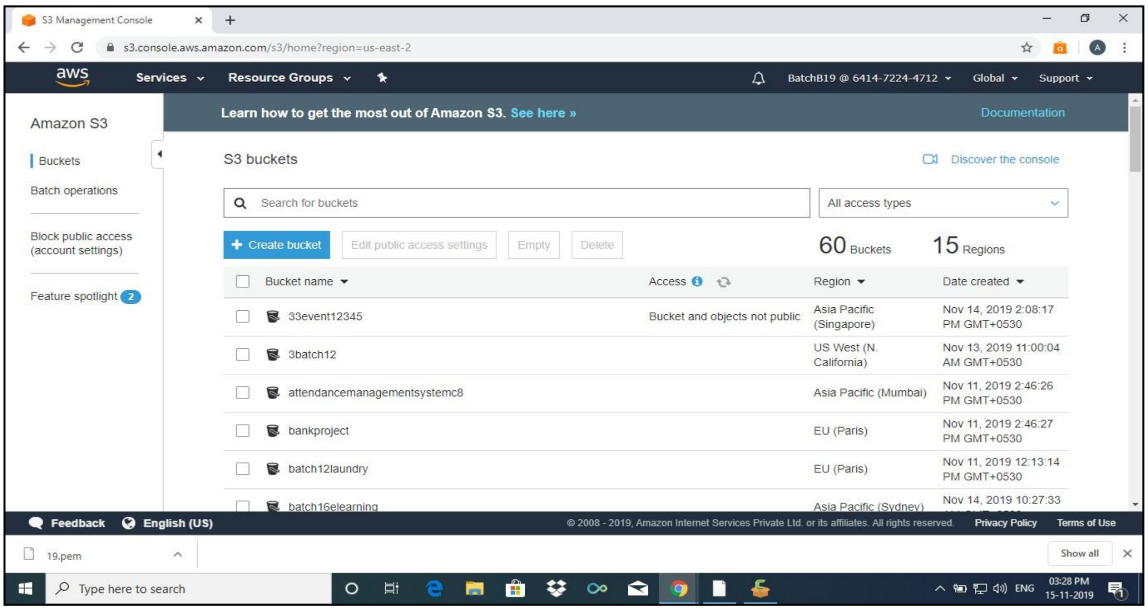
Click **View Instances** to monitor your instances' status. Once your instances are in the **running** state, you can **connect** to them from the Instances screen. Find out how to connect to your instances.

Here are some helpful resources to get you started
- How to connect to your Linux instance
- Learn about AWS Free Usage Tier
- Amazon EC2: User Guide
- Amazon EC2: Discussion Forum

Fig6.2.1.6: Launched Instance

6.2.2 Creating a S3 Bucket in AWS

Step 1: Click on "Create Bucket"


Learn how to get the most out of Amazon S3. See here »

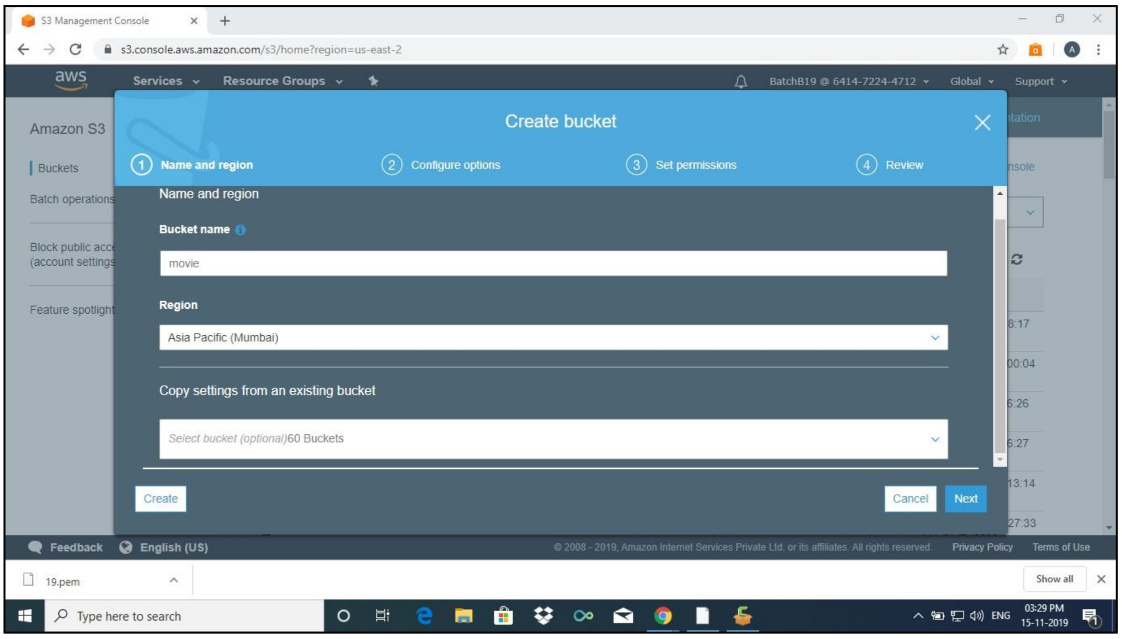
S3 buckets

60 Buckets 15 Regions

Bucket name	Access	Region	Date created
33event12345	Bucket and objects not public	Asia Pacific (Singapore)	Nov 14, 2019 2:08:17 PM GMT+0530
3batch12		US West (N. California)	Nov 13, 2019 11:00:04 AM GMT+0530
attendancemanagementsystemc8		Asia Pacific (Mumbai)	Nov 11, 2019 2:46:26 PM GMT+0530
bankproject		EU (Paris)	Nov 11, 2019 2:46:27 PM GMT+0530
batch12laundry		EU (Paris)	Nov 11, 2019 12:13:14 PM GMT+0530
batch16elearning		Asia Pacific (Sydney)	Nov 14, 2019 10:27:33

Fig 6.2.2.1: Creation of Bucket

Step 2: Select the Name & Region of your Bucket Storage



Name and region

Bucket name: movie

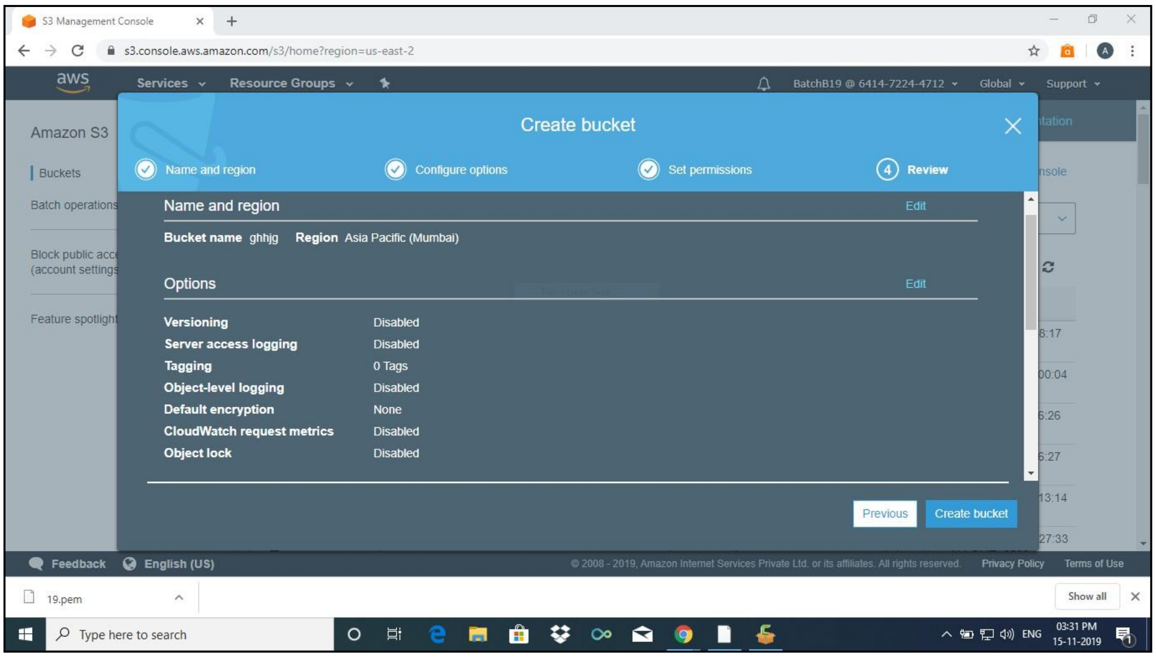
Region: Asia Pacific (Mumbai)

Copy settings from an existing bucket

Select bucket (optional)60 Buckets

Fig 6.2.2.2: Name & Region of the Bucket

Step 3: After Configuring Options & setting Permissions, Review and Click on "Create Bucket"



Name and region

Bucket name ghhjg **Region** Asia Pacific (Mumbai)

Options

Versioning	Disabled
Server access logging	Disabled
Tagging	0 Tags
Object-level logging	Disabled
Default encryption	None
CloudWatch request metrics	Disabled
Object lock	Disabled

Fig 6.2.2.3: Reviewing the Options

Step 4: After creation of Bucket, Upload the Zip File. Click on "Upload"

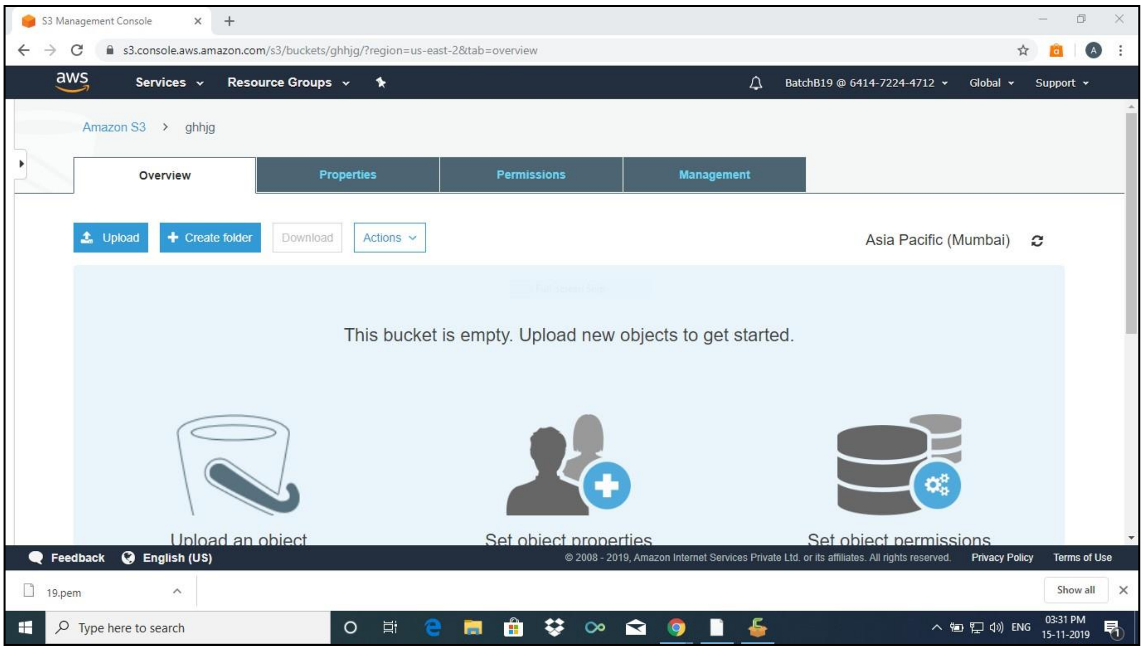


Fig 6.2.2.4: Upload the file

Step 5: Add the file(s) you want to upload

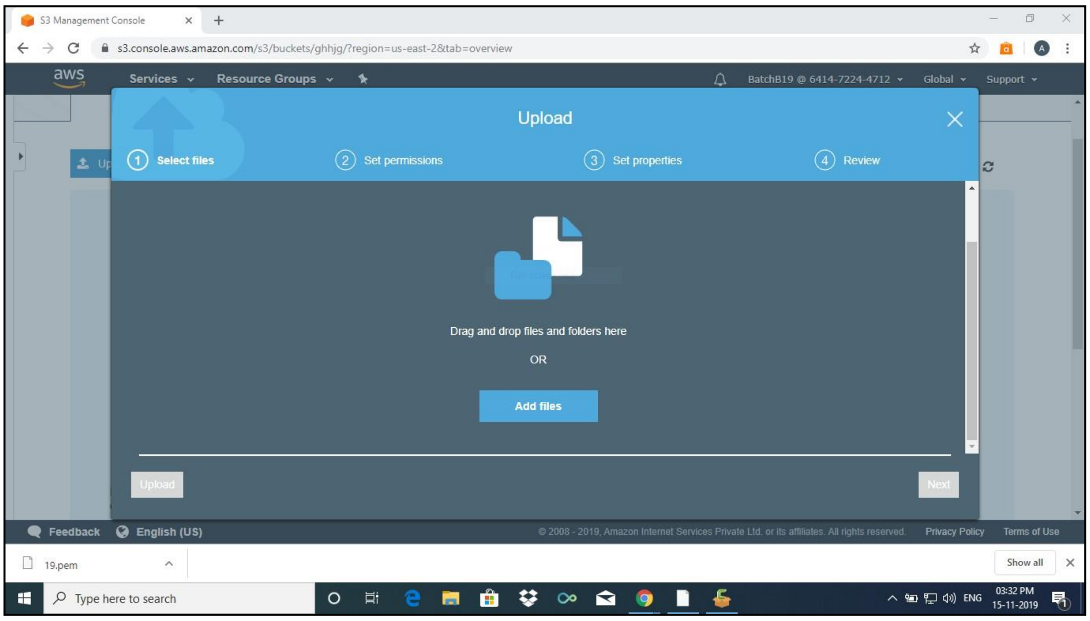
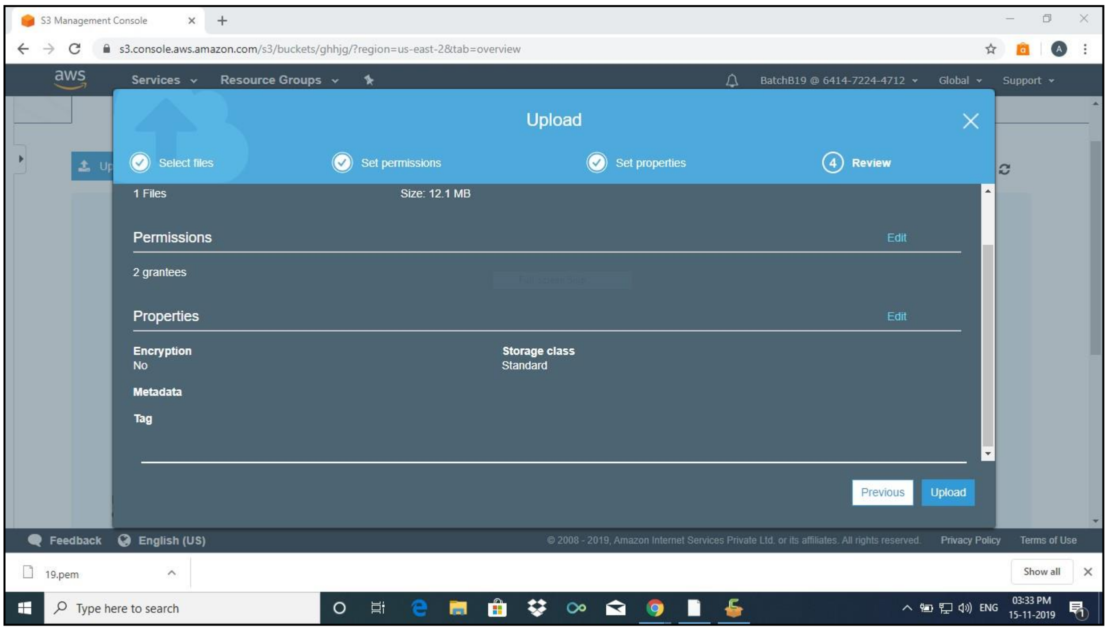

S3 Management Console
s3.console.aws.amazon.com/s3/buckets/ghhjg/?region=us-east-2&tab=overview
Upload
1 Select files
2 Set permissions
3 Set properties
4 Review
Drag and drop files and folders here
OR
Add files
Upload
Next

Fig 6.2.2.5: Add files

Step 6: Then, set permissions and properties and Review. Click on Upload



1 Files Size: 12.1 MB

Permissions

2 grantees

Properties

Encryption
No

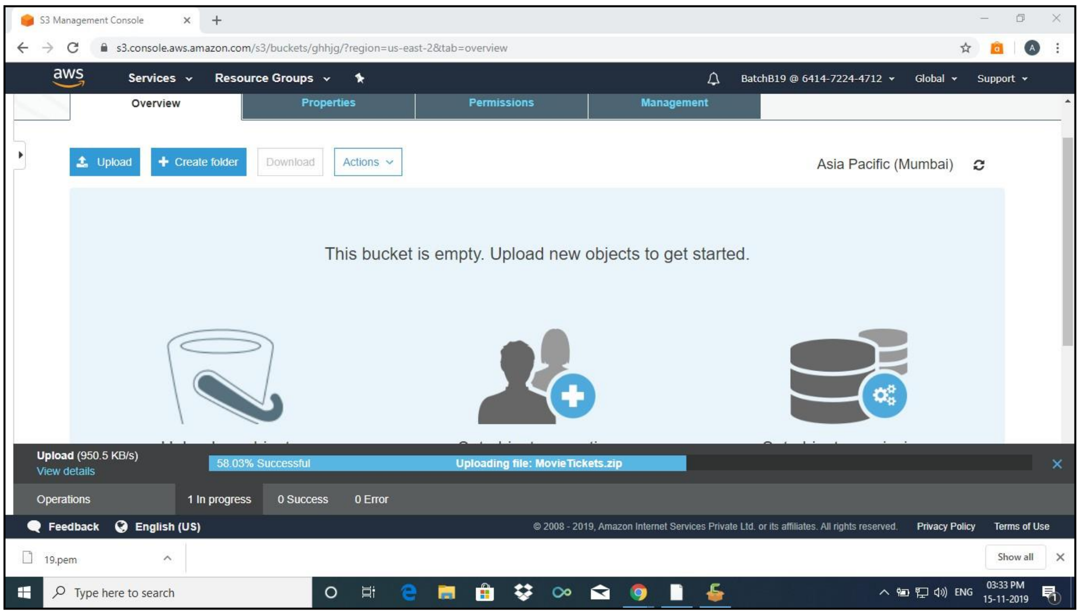
Storage class
Standard

Metadata

Tag

Fig 6.2.2.6: Reviewing Properties

Step 7: Upload will be completed in a few seconds



This bucket is empty. Upload new objects to get started.

Upload (950.5 KB/s)
View details

58.03% Successful Uploading file: MovieTickets.zip

Operations 1 In progress 0 Success 0 Error

Fig 6.2.2.7: Uploading File

7. SYSTEM TESTING

7.1 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing
- Integration Testing
- User Acceptance Testing
- Output Testing
- Validation Testing

7.1.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually, and the module interfaces are verified for the consistency with design specification. All-important processing path are tested for the expected results. All error handling paths are also tested.

7.1.2 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. **Top Down Integration:** This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner. In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. **Bottom-up Integration** This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:
   - The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
   - A driver (i.e.) the control program for testing is written to coordinate test case input and output.
   - The cluster is tested.
   - Drivers are removed and clusters are combined moving upward in the program structure

   The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

7.1.3 User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

7.1.4 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

7.1.5 Validation Checking

Validation checks are performed on the following fields:

Text Field: The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field: The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it must perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data: Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data: Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves. It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true system test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data: Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program. The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

7.2 USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose, the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is quite easy.

7.3 MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing are simple and easy to understand which will make maintenance easier.

8. CONCLUSION

We proposed a novel method to accurately detect malicious social bots in online social networks. Experiments showed that transition probability between user click streams based on the social situation analytics can be used to detect malicious social bots in online social platforms accurately. In future research, additional behaviors of malicious social bots will be further considered, and the proposed detection approach will be extended and optimized to identify specific intentions and purposes of a broader range of malicious social bots.

9. FUTURE ENHANCEMENTS

Future work should be devoted to the development of detection methods that would find their use within operational networks. These methods should rely on the principles of network traffic analysis that would encompass targeted botnet network characteristics and would carry the context that is understandable to the operator of the system. Furthermore, one of the important goals of future detection systems is to operate in real-time thus facilitating timely detection. The future methods should be evaluated using an extensive set of network traces originating from different types of botnets. Finally, special attention should be placed on minimizing the number of errors in identifying botnet network traffic so the proposed methods would performance-wise be suitable for being used in operational networks.

10. REFERENCES

1) F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, ``A new approach to bot detection: Striking the balance between precision and recall," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, San Francisco, CA, USA, Aug. 2016, pp. 533_540.
2) C. A. De Lima Salge and N. Berente, ``Is that social bot behaving unethically?' *Commun. ACM*, vol. 60, no. 9, pp. 29_31, Sep. 2017.
3) M. Sahlabadi, R. C. Muniyandi, and Z. Shukur, ``Detecting abnormal behavior in social networkWebsites by using a process mining technique," *J. Comput. Sci.*, vol. 10, no. 3, pp. 393_402, 2014.
4) F. Brito, I. Petiz, P. Salvador, A. Nogueira, and E. Rocha, ``Detecting social-network bots based on multiscale behavioral analysis," in *Proc. 7th Int. Conf. Emerg. Secur. Inf., Syst. Technol. (SECURWARE)*, Barcelona, Spain, 2013, pp. 81_85.
5) T.-K. Huang, M. S. Rahman, H. V. Madhyastha, M. Faloutsos, and B. Ribeiro, ``An analysis of socware cascades in online social networks," in *Proc. 22nd Int. Conf. World Wide Web*, Rio de Janeiro, Brazil, 2013, pp. 619_630.
6) H. Gao *et al.*, ``Spam ain't as diverse as it seems: Throttling OSN spam with templates underneath," in *Proc. 30th ACSAC*, New Orleans, LA, USA, 2014, pp. 76_85.
7) E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, ``The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96_104, Jul. 2016.
8) T. Hwang, I. Pearce, and M. Nanis, ``Socialbots: Voices from the fronts," *Interactions*, vol. 19, no. 2, pp. 38_45, Mar. 2012.
9) Y. Zhou *et al.*, ``*ProGuard*: Detecting malicious accounts in social network based online promotions," *IEEE Access*, vol. 5, pp. 1990_1999, 2017.
10) Z. Zhang, C. Li, B. B. Gupta, and D. Niu, ``Ef_cient compressed ciphertext length scheme using multi-authority CP-ABE for hierarchical attributes," *IEEE Access*, vol. 6, pp. 38273_38284, 2018. doi: 10.1109/ACCESS.2018.2854600.