**Mining Users Trust from E-Commerce Reviews Based on Sentiment Similarity Analysis**

**ABSTRACT**

Consumers' reviews in E-commerce systems are usually treated as the important resources that reflect user's experience, feelings, and willingness to purchase items. All this information may involve consumers' views on things that can express interest, sentiments, and opinions. Many kinds of research have shown that people are more likely to trust each other with the same attitude toward similar things. In this paper, we consider seeking and accepting sentiments and suggestions in E-commerce systems somewhat implies a form of trust between consumers during shopping. Following this view of point, an E-commerce system reviews mining oriented sentiment similarity analysis approach is put forward to exploring users' similarity and their trust. We divide the trust into two categories, namely direct trust, and propagation of trust, which represents a trust relationship between two individuals. The direct trust degree is obtained from sentiment similarity, and we present an entity-sentiment word pair mining method for similarity feature extraction. The propagation of trust is calculated according to the transitivity feature. Using the proposed trust representation model, we use the shortest path to describe the tightness of trust and put forward an improved shortest path algorithm to figure out the propagation trust relationship between users. A large-scale E-commerce website reviews dataset is collected to examine the accuracy of the algorithms and feasibility of the models. The experimental results indicate that the sentiment similarity analysis can be an efficient method to find trust between users in E-commerce systems.

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. OBJECTIVE   3. EXISTING SYSTEM   1.3.1 EXISTING SYSTEM DISADVANTAGES  1.3.2 LITERATURE SURVEY  1.4 PROPOSED SYSTEM  1.4.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 : PROJECT DESCRIPTION**  2.1 GENERAL  2.2 PROBLEM DEFINATION  2.3 METHODOLOGIES  2.3.1 MODULES NAME  2.3.2 MODULES EXPLANATION  2.3.3 MODULE DIAGRAM  2.3.4 GIVEN INPUTAND EXPECTED OUTPUT  2.4 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS  3.4 Functional Specification  3.5 Non-Functional SpecificatioN |  |
| 4. | **CHAPTER 4 : SYSTEM DESIGN**  4.1 general  4.1.1 USE CASE diagram  4.1.2 CLASS DIAGRAM  4.1.3 OBJECT DIAGRAM  4.1.4 STATE DIAGRAM  4.1.5 ACTIVITY DIAGRAM  4.1.6 SEQUENCED DIAGRAM  4.1.7 COLLABARATION DIAGRAM  4.1.8 COMPONENT DIAGRAM  4.1.9 DEPLOYMENT DIAGRAM  4.1.10 E-R DIAGRAM  4..2 DATA FLOW DIAGRAM  4.2.1 SYSTEM ARCHETECTURE |  |
| 5. | **CHAPTER 5 :SOFTWARE SPECIFICATION**  5.1 general FRONT END  5.2 features of java  5.2.1 the java framework  5.2.2 objective of java  5.2.3 componets of java frmework  5.2.4 Evolution of collection framework  5.3 features of the common language runtime  5.4 j2ee  5.4.1 EVALUTION OF J2EE  5.4.2 THE J2EE DATA ARCHITECTURE  5.5.3 COMPONENT CLASSES THAT MAKE UP  DATA PROVIDERS |  |
| 6. | **CHAPTER 6 : IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION  6.3 DATA BASE TABLE STRUCTURE |  |
| 7. | **CHAPTER 7 : SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 : SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 : APPLICATIONS AND FUTURE ENHANCEMENT**  8.1 GENERAL  8.2 APPLICATIONS  8.3 FUTURE ENHANCEMENTS |  |
| 10. | **CHAPTER 10 :**  9.1 CONCLUSION  9.2 REFERENCES |  |

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO.** |
| 2.3.2 | Module Diagram |  |
| 4.2 | Use case Diagram |  |
| 4.3 | Class Diagram |  |
| 4.4 | Object Diagram |  |
| 4.5 | State Diagram |  |
| 4.6 | Sequence Diagram |  |
| 4.7 | Collaboration Diagram |  |
| 4.8 | Activity Diagram |  |
| 4.9 | Component Diagram |  |
| 4.10 | Data flow Diagram |  |
| 4.11 | E-R Diagram |  |
| 4.12 | Deployment Diagram |  |
| 4.13 | System Architecture |  |
| 7.1 | Home Page |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **NAME** | **NOTATION** | | **DESCRIPTION** | |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | | Represents a collection of similar entities grouped together. | |
| 2. | Association | name  Class B  Class A  Class A  Class B | | Associations represents static relationships between classes. Roles represents the way the two classes see each other. | |
| 3. | Actor | Class A  Class A  Class B  Class B | | It aggregates several classes into a single classes. | |
| 5. | Aggregation | Interaction between the system and external environment | |
| 5. | Relation  (uses) | | Uses | | Used for additional process communication. | |
| 6. | Relation  (extends) | | extends | | Extends relationship is used when one use case is similar to another use case but does a bit more. | |
| 7. | Communication | |  | | Communication between various use cases. | |
| 8. | State | | State | | State of the process. | |
| 9. | Initial State | |  | | Initial state of the object | |
| 10. | Final state | |  | | Final state of the object | |
| 11. | Control flow | |  | | Represents various control flow between the states. | |
| 12. | Decision box | |  | | Represents decision making process from a constraint | |
| 13. | Usecase | |  | | Interact ion between the system and external environment. | |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which is a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors, etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |
| 20. | Message | Message | Represents the message exchanged. |

**LIST OF ABBREVATION**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVATION** | **EXPANSION** |
| 1**.** | DB | DataBase |
| 2. | JVM | Java Virtual Machine |
| 3. | JSP | Java Server Page |
| 4. | CB | Collective Behavior |
| **5.** | RSSS | Ramp secret sharing scheme |
| 6. | JRE | Java Runtime Environment |

**CHAPTER 1**

**INTRODUCTION**

**1.1 GENERAL:**

Reviews from consumers are very important information in E-commerce systems. Many online shops have developed reviews system for users to post their reviews. With the rapid development of social networking media, more and more people are willing to share their feelings, opinions and suggestions on their bought items with their friends or even strangers in social network applications or E-commerce systems. These reviews can be very useful for people's decision making in many different scenarios such as users' preference mining and personalized recommendation [1], [2]. At present, more and more review mining based applications are being applied to make our decision process easier than before.

These applications have greatly changed people's behaviour patterns, especially in E-commerce activities. For example, when people want to buy a product, book a hotel or restaurant, they usually not only ask for advice from their friends but also refer to reviews available online. To adapt to this change, many famous E-commerce companies, such as Amazon, eBay and Taobao (China), have built up well-function consumer review systems.

**1.2. OBJECTIVE**

Using the proposed trust representation model, we use the shortest path to describe the tightness of trust and put forward an improved shortest path algorithm to figure out the propagation trust relationship between users. A large-scale E-commerce website reviews dataset is collected to examine the accuracy of the algorithms and feasibility of the models. The experimental results indicate that the sentiment similarity analysis can be an efficient method to find trust between users in E-commerce systems.

* 1. **Existing System:**
* Consumers' reviews in E-commerce systems are usually treated as the important resources that reflect user's experience, feelings, and willingness to purchase items. All this information may involve consumers' views on things that can express interest, sentiments, and opinions.
* Many kinds of research have shown that people are more likely to trust each other with the same attitude toward similar things.
* It is impossible for a consumer to judge whose reviews are suitable and which users can be trusted.

**1.3.1 Existing System Disadvantages:**

* Can’t distinguish from large database.
* In accurate sentiment analysis

**1.3.2 LITERATURE SURVEY**

**TITLE :** TruCom: Exploiting domain-specific trust networks for multicategory item recommendation

**AUTHOR :** H. Liu, F. Xia, Z. Chen, N. Y. Asabere, J. Ma, and R. Huang

**YEAR :** 2017

**DESCRIPTION**

Recommender systems (RSs) have become important tools for solving the problem of information overload. With the advent and popularity of online social networks, some studies on network-based recommendation have emerged, raising the concern of many researchers. Trust is one kind of important information available in social networks and is often used for performance improvement in social-network-based RSs. However, most trust-aware RSs ignore the fact that people trust different subsets of friends pertaining to different domains, such as music and movies, because people behave differently in diverse domains according to different interests. This paper proposes a novel recommendation method called TruCom. In a multicategory item recommendation domain, TruCom first generates a domain-specific trust network pertaining to each domain and then builds a unified objective function for improving recommendation accuracy by incorporating the hybrid information of direct and indirect trust into a matrix factorization recommendation model. Through relevant benchmark experiments on two real-world data sets, we show that TruCom achieves better performance than other existing recommendation methods, which demonstrates the effectiveness and reliability of TruCom.

**TITLE :** Effects of sentiment on recommendations in social network

**AUTHOR :** P.-Y. Hsu, H.-T. Lei, S.-H. Huang, T. H. Liao, Y.-C. Lo, and C.-C. Lo

**YEAR :** 2018

**DESCRIPTION**

This study adopted a sentiment word database to extract sentiment-related data from microblog posts. These data were then used to investigate the effect of different types of sentiment-related words on product recommendations. The results indicate that posts containing strong sentiments received more clicks than posts containing neutral sentiments. Posts containing more than one positive sentiment word generate more effective recommendations than posts containing only one positive sentiment word. This study also demonstrated that posts with a negative polarity classification received more clicks than those with a positive polarity classification. Additionally, the microblog posts containing implicit sentiment words received more clicks than those containing explicit sentiment words. The findings presented here could assist product or service marketers who use Plurk or similar microblogging platforms better focus their limited financial resources on potential online customers to achieve maximum sale revenue.

**TITLE :** The joint beta distribution with refund rate in online C2C trust building: A theoretical study on Taobao

**AUTHOR :** C. Qin, W. Siyi, and A. Lin

**YEAR :** 2012

**DESCRIPTION**

Lack of trust is one of the fundamental reasons for losing customers from “faceless” e-commerce websites in the consumer-to-consumer market. In order to combat problems with dishonest market participants, a reputation system based on the trustworthiness of sellers has been widely established by the service provider. Traditionally this trust model is mainly based on feedback mechanism while neglecting the other important factors, such as, refund rate, which may lead to transaction risks in the new transaction. This paper proposes a global reputation rating method by using joint beta probability density functions to combine positive feedback rating and refund rate. The new trust model has the advantage of tractability and scalability as well as its theoretical sound basis on statistics.

**TITLE :** A robust reputation-based computational model for trust establishment in pervasive systems

**AUTHOR :** S. Kraounakis, I. N. Demetropoulos, A. Michalas, M. S. Obaidat, P. G. Sarigiannidis, and M. D. Louta

**YEAR :** 2015

**DESCRIPTION**

Distributed systems built in open competitive and highly dynamic pervasive environments are composed of autonomous entities that act and interact in an intelligent and flexible manner so as to achieve their own goals and aims. System entities may be classified into two main categories that are, in principle, in conflict. These are the service resource requestors (SRRs) wishing to use services and/or exploit resources offered by the other system entities and the service resource providers (SRPs) that offer the services/resources requested. Seeking for the maximization of their welfare, entities may misbehave, thus leading to a significant deterioration of system's performance. The scope of this paper is to present a computational model for trust establishment based on a reputation mechanism, which incorporates direct SRRs' experiences and information disseminated from witness SRRs on the basis of their past experiences with SRPs. The designed mechanism discriminates between unfair feedback ratings intentionally and unintentionally provided, takes into consideration potential changes to providers' behavior, and weighs more recent events in the evaluation of the overall reputation ratings. The proposed model has been extensively evaluated through simulation experiments. It exhibits good performance, as the reputation computation error introduced due to false feedback provision decreases significantly.

**TITLE :** Trust and compactness in social network groups

**AUTHOR :** P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné

**YEAR :** 2015

**DESCRIPTION**

Understanding the dynamics behind group formation and evolution in social networks is considered an instrumental milestone to better describe how individuals gather and form communities, how they enjoy and share the platform contents, how they are driven by their preferences/tastes, and how their behaviors are influenced by peers. In this context, the notion of compactness of a social group is particularly relevant. While the literature usually refers to compactness as a measure to merely determine how much members of a group are similar among each other, we argue that the mutual trustworthiness between the members should be considered as an important factor in defining such a term. In fact, trust has profound effects on the dynamics of group formation and their evolution: individuals are more likely to join with and stay in a group if they can trust other group members. In this paper, we propose a quantitative measure of group compactness that takes into account both the similarity and the trustworthiness among users, and we present an algorithm to optimize such a measure. We provide empirical results, obtained from the real social networks EPINIONS and CIAO, that compare our notion of compactness versus the traditional notion of user similarity, clearly proving the advantages of our approach.

**TITLE :** Trust based recommendation systems

**AUTHOR :** M. G. Ozsoy and F. Polat

**YEAR :** 2013

**DESCRIPTION**

Internet of Things (IoT) creates a world where smart objects and services interacting autonomously. Taking into account the dynamic-heterogeneous characteristic of interconnected devices in IoT, demand for a trust model to guarantee security, authentication, authorization, and confidentiality of connected things, regardless of their functionality, is imperative. However, as far as we know, against the centrality of trust-based recommendation mechanisms in the IoT environment, there is no ambient study for investigating its techniques. In this paper, we present a systematic literature review (SLR) of trust based IoT recommendation techniques so far. Detailed classifications based on extracted parameters as well as investigation existing techniques in three different IoT layers put forth. Moreover, the advantages, disadvantages and open issues of each approach are introduced that can expand more frontier in obtaining accurate IoT recommendation in the future.

* 1. **Proposed System**
* In this paper, we consider seeking and accepting sentiments and suggestions in E-commerce systems somewhat implies a form of trust between consumers during shopping.
* Following this view of point, an E-commerce system reviews mining oriented sentiment similarity analysis approach is put forward to exploring users' similarity and their trust.

**1.4.1 Proposed System Advantages**

* Sentiment similarity of reviews has increased.
* Propagation of users trust relation has increased with performance.

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

Due to its human-related properties, trust is difficult to be uniformly defined or even precisely described. The vast majority of existing studies focused on trust construction and maintenance between customers and companies over time and after repeated experiences. While limited effort is spent on trust between consumers and potential consumers in E-commerce systems. Obviously, in the field of E-commerce reviews, people are more concerned about the credibility of reviews and the trust of user who post the reviews. In our work, we aim to investigate trust between users in E-commerce systems quantitatively by exploring their reviews and evaluations regarding to various commodities, services, businesses, and other related subjects.

Based on sentiment analysis of large-scale text reviews in E-commerce website, we focus on sentiment similarity between users to establish their trust, which can provide potential support for further implementation of trust related recommendation service.

**2.3 METHODOLOGIES**

**2.3.1** **MODULES NAME:**

**This project having the following five modules:**

**1. User Interface Design**

**2. Admin**

**3. Sentiment Similarity Analysis**

**4. User**

**5. Entity-Sentiment Word Pairs Extraction**

1. **User Interface Design**

In this module we design the windows for the project. These windows are used for secure login for all users. To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page.

User Login

Server

Database

Home Page

Register& Login Page

1. **Admin**

Admin is the second module in this project, where he will be playing a very crucial role in the entire

**Admin:** His roles and responsibilities in the project are given below in bullet points.

**View Authorized users:** Where he can see all the registered users.

**Add and View Category:** He will be adding the products to be sold according to the category.

**View All Products with Ratings:** All the user purchased products with rating can be seen here

**View All Product Reviews:** All the user purchased products with reviews can be seen here

**View All Statement Similarity by Reviews:** Product rating & review according to the sentiment similarity will be sorted here.

**View Purchased Products:** All products purchased by the users can be seen here.

**View User Query Keyword:** Most searched keywords by the users will be displayed here for evaluation purpose.

Admin

View Authorized users

Add and View Category

Remove UsView All Product Reviews

View All Products with Ratings

View All Statement Similarity By Reviews

Database

View Purchased Products&User Query Keyword

1. **Sentiment Similarity Analysis:**

The methods take the similarity analysis as an important and basic content, which consider the sentiment and emotion as the evaluation factors for trust. Additionally, sentiment and affective similarity analysis have been studied extensively in natural language understanding, data mining and statistical analysis.

Admin

Natural Language Understanding

Data Mining

Statistical Analysis

Result

Database

1. **User**

In this module all the operation will be done by the user. User has following operations displayed below.

**Profile:** He can view his profile, and if wants he can update his profile also.

**Search Friends:** In this module we will be giving a search box for the user where he will be able to search his friends. (If they are registered in the site).

**View Friend Request:** If any one sends a request to other users present in the site they can see the status here.

**View My Friends:** All users friends can be viewed here.

**Delete My Friends:** user can delete their friends in this module.

**Search Products and Buy:** User can search the products which were uploaded by the admin and can purchase them.

**View Purchases:** All the user purchased goods can be viewed here & he can give review and rating for those products where sentiment similarity will be calculated.

**Friend Suggestions:** If any friends suggests him any product he can see them here.

User

View Authorized Users Profile

Search Friends

View My Friends & Delete My Friends

View Friend Request

Search Products, Buy & View Purchases

Database

Friend Suggestions

1. **Entity-Sentiment Word Pairs Extraction:**

In entity-sentiment word pair’s extraction, the entities usually are nouns or noun phrases which represent some specie objects, features, or attributes, etc. The sentiments are adjectives or adverbs which express emotions, opinions, or tendencies, etc. We apply the association rules to extract frequently occurring nouns or noun phrases as entities, and we use the adjectives or adverbs as sentiment which have the closest information distance to the object.

Admin

Emotions

Opinions

Tendencies

Result

Database

**GIVEN INPUT EXPECTED OUTPUT:**

* **User Interface Design**

Input : Enter Login name and Password

Output : If valid user name and password then directly open the home page otherwise show error message and redirect to the registration page.

* **Admin**

Input : Admin Login name and Password

Output: If valid user name and password then directly open the admin home page otherwise show error message and redirect to the admin login page.

* **Sentiment Similarity Analysis**

Input : Admin verify all data

Output : Verifying all sentiment similarity analysis process.

* **User**

Input : Enter the user name and password

Output: If valid user name and password then directly open the user home page otherwise show error message and redirect to the user login page.

* **Entity-Sentiment Word Pairs Extraction:**

Input : Verifying Entity-Sentiment Word Pairs Extraction.

Output: In entity-sentiment word pairs extraction, the entities usually are nouns or noun phrases which represent some specie objects, features, or attributes, etc.

* 1. **TECHNIQUE USED OR ALGORITHM USED**

**Propagation Trust Computing Protocol**

Propagation trust of two users can be acquired from direct trust between intermediate users. Any two user nodes can be acquired by adding up direct trust weight between all nodes in the shortest trust path.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

The methods take the similarity analysis as an important and basic content, which consider the sentiment and emotion as the evaluation factors for trust. Additionally, sentiment and affective similarity analysis have been studied extensively in natural language understanding, data mining and statistical analysis

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shoulds what the system do and not how it should be implemented.

**HARDWARE**

* PROCESSOR : PENTIUM IV 2.6 GHz, Intel Core 2 Duo.
* RAM : 512 MB DD RAM
* MONITOR : 15” COLOR
* HARD DISK : 40 GB

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* + Front End : J2EE (JSP, SERVLET)
  + Back End : MY SQL 5.5
  + Operating System : Windows 7
  + IDE : Eclipse

**3.4 FUNCTIONAL REQUIREMENTS**

A functional requirement defines a function of a software-system or its component. A function is described as a set of inputs, the behaviour, and outputs. The outsourced computation is data is more secured.

* **User Interface**
* **Admin:**
* View Authorized Users:
* Add and View Category
* View All Products with Ratings
* View All Product Reviews
* View All Statement Similarity By Reviews
* View Purchased Products
* View User Query Keyword
* **Sentiment Similarity Analysis**
* **User:**
* Profile
* Search Friends
* View Friend Request
* View My Friends.
* Delete My Friends
* Search Products and Buy
* View Purchases
* Friend Suggestions
* **Entity-Sentiment Word Pairs Extraction**

**3.5 NON-FUNCTIONAL REQUIREMENTS**

The major non-functional Requirements of the system are as follows

* **Usability**

The system is designed with completely automated process hence there is no or less user intervention.

* **Reliability**

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

* **Performance**

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

* **Supportability**

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

* **Implementation**

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform. Interface the user interface is based on Struts provides HTML Tag

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

**4.2 Use Case Diagram**



**EXPLANATION:**

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. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.3 Class Diagram**



**EXPLANATION**

In this class diagram represents how the classes with attributes and methods are linked together to perform the verification with security. From the above diagram shown the various classes involved in our project

**4.4 Object Diagram**



**EXPLANATION:**

In the above digram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.5 State Chart Diagram**



**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.6 Sequence Diagram**



**EXPLANATION:**

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. 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.

**4.7 Collaboration Diagram**

****

**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.8 Activity Diagram**



**EXPLANATION:**

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

**4.9 Component Diagram**



**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.10 Data Flow Diagram:**

**Level 0:**

Register

Home Page

Login

Verify Details

Data base

Error page

**Level 1:**

User

View Authorized Users Profile

Search Friends

View My Friends & Delete My Friends

View Friend Request

Search Products, Buy & View Purchases

Database

Friend Suggestions

**Level 2:**

Admin

Natural Language Understanding

Data Mining

Statistical Analysis

Result

Database

**EXPLANATION:**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

**4.11 E-R Diagram:**

Database

Login

Register

Search

Keywords

Products

Ranking & Reviews

Login

Add Products

View Data

Verify

View All Statement Similarity By Reviews

Ranking & Reviews

**EXPLANATION:**

Entity-Relationship Model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database.

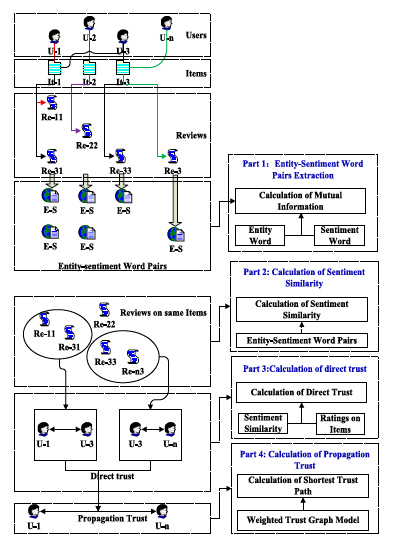
**4.12 Deployment Diagram:**



**EXPLANATION:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger deployment and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination. Results are to be showed to user by data aggregators. All boxes are arrow indicates dependencies.

**4.13 System Architecture**

****

**EXPLANATION:**

To find the trust, including direct and propagation, based on sentiment similarity of reviews by users in E-commerce the relationship between each entity word and the sentiment word. And then we can find those words pairs with close relationship.

Firstly, the entity-sentiment word pairs are extracted from reviews. The step is a key process to further deal with the sentiment similarity analysis and direct trust computing. The extraction of entity-sentiment word pairs in part 1 is mainly to analyses the vocabulary of the text, extract the entity words and sentiment words which describing the object. We use NLProcessor linguistic parser for entity word and high frequency word combined with public lexicon for sentiment word. A mutual information formula is used to calculate the relationship between each entity word and the sentiment word. And then we can find those words pairs with close relationship.

Secondly, we perform sentiment similarity calculations for two user-related reviews based on entity-sentiment word pairs. Calculation of sentiment similarity in part 2 is to compute the similarity degree of different reviews texts. This step is to use the obtained entity-sentiment word pairs for comparative analysis.

Thirdly, we use a new formula to calculate direct trust between two users with a common review objects. Calculation of direct trust in part 3 is to compute direct trust whom have reviews on same item or object. The calculation method mainly includes two aspects, one is sentiment similarity, and the other is the user's rating of the object.

Finally, calculation of propagation trust in part 4, assume users as nodes, and the links, which are based on their direct trust, as edges to create a trust network. Then we use an improved shortest path algorithm to calculate the propagation trust links between each pair of user nodes

**CHAPTER 5**

**DEVELOPMENT TOOLS**

* 1. **GENERAL**

This chapter is about the software language and the tools used in the development of the project. The platform used here is JAVA. The Primary languages are JAVA, J2EE and J2ME. In this project J2EE is chosen for implementation.

**5.2 FEATURES OF JAVA**

**5.2.1 THE JAVA FRAMEWORK**

**Java** is a programming language originally developed by James Gosling at Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications are typically compiled to byte code that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is general-purpose, concurrent, class-based, and object-oriented, and is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

Java is considered by many as one of the most influential programming languages of the 20th century, and is widely used from application software to web applications the java framework is a new platform independent that simplifies application development internet. Java technology's versatility, efficiency, platform portability, and security make it the ideal technology for network computing. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

**5.2.2 OBJECTIVES OF JAVA**

To see places of Java in Action in our daily life, explore java.com.

## Why Software Developers Choose Java

Java has been tested, refined, extended, and proven by a dedicated community. And numbering more than 6.5 million developers, it's the largest and most active on the planet. With its versatility, efficiency, and portability, Java has become invaluable to developers by enabling them to:

* Write software on one platform and run it on virtually any other platform
* Create programs to run within a Web browser and Web services
* Develop server-side applications for online forums, stores, polls, HTML forms processing, and more
* Combine applications or services using the Java language to create highly customized applications or services
* Write powerful and efficient applications for mobile phones, remote processors, low-cost consumer products, and practically any other device with a digital heartbeat

## Some Ways Software Developers Learn Java

Today, many colleges and universities offer courses in programming for the Java platform. In addition, developers can also enhance their Java programming skills by reading Sun's java.sun.com Web site, subscribing to Java technology-focused newsletters, using the Java Tutorial and the New to Java Programming Center, and signing up for Web, virtual, or instructor-led courses.

**ObjectOriented** To be an Object Oriented language, any language must follow at least the four characteristics.

1. Inheritance   :It is the process of creating the new classes and using the behavior of the existing classes by extending them just to reuse  the existing code and adding addition a features as needed.

2. Encapsulation: It is the mechanism of combining the information and providing the abstraction.

3. Polymorphism: As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the  methods.

4. Dynamic binding: Sometimes we don't have the knowledge of objects about their specific types while writing our code. It is the way of providing the maximum functionality to a program about the specific type at runtime.

**5.2.3 JAVA SWING OVERVIEW**

**Abstract Window Toolkit (AWT) is cross-platform**

Swing[[1]](http://en.wikibooks.org/wiki/Java_Programming/Swing#cite_note-0) provides many controls and widgets to build user interfaces with. Swing class names typically begin with a J such as JButton, JList, JFrame. This is mainly to differentiate them from their AWT counterparts and in general is one-to-one replacements. Swing is built on the concept of Lightweight components vs AWT and SWT's concept of Heavyweight components. The difference between the two is that the Lightweight components are rendered (drawn) using purely Java code, such as drawLine and drawImage, whereas Heavyweight components use the native operating system to render the components.

Some components in Swing are actually heavyweight components. The top-level classes and any derived from them are heavyweight as they extend the AWT versions. This is needed because at the root of the UI, the parent windows need to be provided by the OS. These top-level classes include JWindow, JFrame, JDialog and JApplet. All Swing components to be rendered to the screen must be able to trace their way to a root window of one of those classes.

**Note**: It generally it is not a good idea to mix heavyweight components with lightweight components (other than as previously mentioned) as you will encounter layering issues, e.g., a lightweight component that should appear "on top" ends up being obscured by a heavyweight component. The few exceptions to this include using heavyweight components as the root pane and for popup windows. Generally speaking, heavyweight components will render on top of lightweight components and will not be consistent with the look and feel being used in Swing. There are exceptions, but that is an advanced topic. The truly adventurous may want to consider reading this [article](http://java.sun.com/products/jfc/tsc/articles/mixing/) from Sun on mixing heavyweight and lightweight components.

**5.2.4 Evolution of Collection Framework:**

Almost all collections in Java are derived from the [**java.util.Collection**](http://download.oracle.com/javase/7/docs/api/java/util/Collection.html) interface. Collection defines the basic parts of all collections. The interface states the add() and remove() methods for adding to and removing from a collection respectively. Also required is the toArray() method, which converts the collection into a simple array of all the elements in the collection. Finally, the contains() method checks if a specified element is in the collection. The Collection interface is a sub interface of [**java.util.Iterable**](http://download.oracle.com/javase/7/docs/api/java/util/Iterable.html), so the iterator() method is also provided. All collections have an iterator that goes through all of the elements in the collection. Additionally, Collection is a generic. Any collection can be written to store any class. For example, Collection<String> can hold strings, and the elements from the collection can be used as strings without any casting required.

There are three main types of collections:

* Lists: always ordered, may contain duplicates and can be handled the same way as usual arrays
* Sets: cannot contain duplicates and provide random access to their elements
* Maps: connect unique keys with values, provide random access to its keys and may host duplicate values

**LIST:**

Lists are implemented in the JCF via the java.util.List interface. It defines a list as essentially a more flexible version of an array. Elements have a specific order, and duplicate elements are allowed. Elements can be placed in a specific position. They can also be searched for within the list. Two concrete classes implement List. The first is java.util.ArrayList, which implements the list as an array. Whenever functions specific to a list are required, the class moves the elements around within the array in order to do it. The other implementation is java.util.LinkedList. This class stores the elements in nodes that each have a pointer to the previous and next nodes in the list. The list can be traversed by following the pointers, and elements can be added or removed simply by changing the pointers around to place the node in its proper place.

**SET:**

Java's [java.util.Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html) interface defines the set. A set can't have any duplicate elements in it. Additionally, the set has no set order. As such, elements can't be found by index. Set is implemented by java.util.HashSet,java.util.LinkedHashSet, and java.util.TreeSet. HashSet uses a hash table. More specifically, it uses a [java.util.HashMap](http://download.oracle.com/javase/7/docs/api/java/util/HashMap.html) to store the hashes and elements and to prevent duplicates. Java.util.LinkedHashSet extends this by creating a doubly linked list that links all of the elements by their insertion order. This ensures that the iteration order over the set is predictable. [java.util.TreeSet](http://download.oracle.com/javase/7/docs/api/java/util/TreeSet.html) uses a red-black tree implemented by a [java.util.TreeMap](http://download.oracle.com/javase/7/docs/api/java/util/TreeMap.html). The red-black tree makes sure that there are no duplicates. Additionally, it allows Tree Set to implement java.util.SortedSet.

The [java.util.Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html) interface is extended by the java.util.SortedSet interface. Unlike a regular set, the elements in a sorted set are sorted, either by the element's compareTo() method, or a method provided to the constructor of the sorted set. The first and last elements of the sorted set can be retrieved, and subsets can be created via minimum and maximum values, as well as beginning or ending at the beginning or ending of the sorted set. The SortedSet interface is implemented by java.util.TreeSet

[java.util.SortedSet](http://download.oracle.com/javase/7/docs/api/java/util/SortedSet.html) is extended further via the java.util.NavigableSet interface. It's similar to SortedSet, but there are a few additional methods. The floor(), ceiling(), lower(), and higher() methods find an element in the set that's close to the parameter. Additionally, a descending iterator over the items in the set is provided. As with SortedSet, java.util.TreeSet implements NavigableSet.

**MAP:**

Maps are defined by the java.util.Map interface in Java. Maps are simple data structures that associate a key with a value. The element is the value. This lets the map be very flexible. If the key is the hash code of the element, the map is essentially a set. If it's just an increasing number, it becomes a list. Maps are implemented by java.util.HashMap, java.util.LinkedHashMap, and java.util.TreeMap. HashMap uses a hash table. The hashes of the keys are used to find the values in various buckets. LinkedHashMap extends this by creating a doubly linked list between the elements. This allows the elements to be accessed in the order in which they were inserted into the map. TreeMap, in contrast to HashMap and LinkedHashMap, uses a red-black tree. The keys are used as the values for the nodes in the tree, and the nodes point to the values in the map

**Thread:**

Simply put, a threadis a program's path of execution. Most programs written today run as a single thread, causing problems when multiple events or actions need to occur at the same time. Let's say, for example, a program is not capable of drawing pictures while reading keystrokes. The program must give its full attention to the keyboard input lacking the ability to handle more than one event at a time. The ideal solution to this problem is the seamless execution of two or more sections of a program at the same time.

## Creating threads

Java's creators have graciously designed two ways of creating threads: implementing an interface and extending a class. Extending a class is the way Java inherits methods and variables from a parent class. In this case, one can only extend or inherit from a single parent class. This limitation within Java can be overcome by implementing interfaces, which is the most common way to create threads. (Note that the act of inheriting merely allows the class to be run as a thread. It is up to the class to start() execution, etc.)

Interfaces provide a way for programmers to lay the groundwork of a class. They are used to design the requirements for a set of classes to implement. The interface sets everything up, and the class or classes that implement the interface do all the work. The different set of classes that implement the interface have to follow the same rules.

**5.5 Conclusion**

Swing's high level of flexibility is reflected in its inherent ability to override the native host [operating system](http://en.wikipedia.org/wiki/Operating_system) (OS)'s GUI controls for displaying itself. Swing "paints" its controls using the Java 2D APIs, rather than calling a native user interface toolkit. The Java thread scheduler is very simple. All threads have a priority value which can be changed dynamically by calls to the threads setPriority() method . Implementing the above concepts in our project to do the efficient work among the Server.

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

The Implementation is nothing but sores code of project.

**6.2 IMPLEMENTATION**

**Coding:**

**UserHome.jsp**

<!DOCTYPE HTML>

<html>

<head>

<title>User Page</title>

<meta charset=*"utf-8"* />

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"* />

<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->

<link rel=*"stylesheet"* href=*"assets/css/main.css"* />

<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->

<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->

</head>

<body class=*"landing"*>

<!-- Header -->

<header id=*"header"* class=*"alt"*>

<h1><a>Sentiment Similarity Analysis</a></h1>

<a href=*"#nav"*>Menu</a>

</header>

<!-- Nav -->

<nav id=*"nav"*>

<ul class=*"links"*>

<li><a href=*"userprofile.jsp"*>Profile</a></li>

<li><a href=*"searchFriends.jsp"*>Search Friends</a></li>

<li><a href=*"ViewFriendRequest.jsp"*>View Friend Request</a></li>

<li><a href=*"ViewMyFriends.jsp"*>View My Friends</a></li>

<li><a href=*"DeleteMyFriends.jsp"*>Delete My Friends</a></li>

<li><a href=*"SearchProductsandRecommend.jsp"*>Search Products and Recommend</a></li>

<li><a href=*"ViewPostRecommends.jsp"*>View Post Recommends</a></li>

<li><a href=*"FriendProductConsumers.jsp"*>Friend Product Consumers</a></li>

<li><a href=*"./LogoutServlet"*>Logout</a></li>

</ul>

</nav>

<!-- Banner -->

<section id=*"banner"*>

<i class=*"icon fa-diamond"*></i>

<h2>Mining Users Trust From E-Commerce Reviews Based On Sentiment Similarity Analysis</h2>

</section>

<!-- One -->

<section id=*"two"* class=*"wrapper special"*>

<div class=*"inner"*>

<header class=*"major narrow"*>

<h2><%String s= request.getParameter("status");

**if**(s!=**null**)

{

out.print(s);

}

%></h2>

</header>

<div class=*"image-grid"*>

</div>

<ul class=*"actions"*>

</ul>

</div>

</section>

<!-- Scripts -->

<script src=*"assets/js/jquery.min.js"*></script>

<script src=*"assets/js/skel.min.js"*></script>

<script src=*"assets/js/util.js"*></script>

<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->

<script src=*"assets/js/main.js"*></script>

</body>

</html>

**RegistrationForm.jsp**

<!DOCTYPE HTML>

<html>

<head>

<title>User Registration</title>

<meta charset=*"utf-8"* />

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"* />

<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->

<link rel=*"stylesheet"* href=*"assets/css/main.css"* />

<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->

<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->

<script type=*"text/javascript"*>

**function** valid()

{

**var** x = document.forms["validate"]["uname"].value;

**if**(x=="")

{

alert("username must be filled");

**return** **false**;

}

**var** x = document.forms["validate"]["password"].value;

**if**(x=="")

{

alert("password must be filled")

**return** **false**;

}

**var** y = document.forms["validate"]["cpassword"].value;

**if**(y=="")

{

alert("confirm password must be filled");

**return** **false**;

}

**if**(x!=y)

{

alert("Password and confirm password not matched");

**return** **false**;

}

**var** email = document.forms["validate"]["email"].value;

**if**(email=="")

{

alert("email must be filled");

**return** **false**;

}

**if**(!/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(email))

{

alert("Enter Valid Email");

**return** **false**;

}

**var** mobile = document.forms["validate"]["mobile"].value;

**if**(mobile=="")

{

alert("Mobile Number must be filled");

**return** **false**;

}

**if**(!/^[0-9]{1,10}$/.test(mobile))

{

alert("Mobile Number should be in digits");

**return** **false**;

}

**if**(!/^\d{10}$/.test(mobile))

{

alert("Mobile Number should be 10 digits");

**return** **false**;

}

**var** dob = document.forms["validate"]["dob"].value;

**if**(dob=="")

{

alert("Select Your Date of Birth");

**return** **false**;

}

**var** gender = document.forms["validate"]["gender"].value;

**if**(gender=="")

{

alert("Select gender");

**return** **false**;

}

**var** addr = document.forms["validate"]["addr"].value;

**if**(addr=="")

{

alert("Address Must be filled");

**return** **false**;

}

**var** commerce = document.forms["validate"]["commerce"].value;

**if**(commerce=="")

{

alert("Please Select E-commerce Site");

**return** **false**;

}

}

</script>

</head>

<body class=*"landing"*>

<!-- Header -->

<header id=*"header"* class=*"alt"*>

<h1><a>Sentiment Similarity Analysis</a></h1>

<!--<a href="#nav">Menu</a>

--></header>

<!-- Nav -->

<!--<nav id="nav">

<ul class="links">

<li><a href="LoginForm.jsp">Login</a></li>

</ul>

</nav>

--><!-- Banner -->

<section id=*"banner"*>

<i class=*"icon fa fa-diamond"*></i>

<h2>Mining Users Trust From E-Commerce Reviews Based On Sentiment Similarity Analysis</h2>

<ul class=*"actions"*>

<li><a href=*"RegistrationForm.jsp"* class=*"button big alt"*>SignUp</a> <a href=*"LoginForm.jsp"* class=*"button big alt"*>Login</a></li>

</ul>

</section>

<!-- One -->

<section id=*"two"* class=*"wrapper special"*>

<div class=*"inner"*>

<header class=*"major narrow"*>

<h2 class=*"button big alt"*>Registration Form</h2>

<p>Fill All The Fields</p>

<font style="font-family: *serif*;">

<%String s = request.getParameter("status");

**if**(s!=**null**)

{

out.print(s);

}

%>

</font>

</header>

<div class=*"actions"*>

<form action=*"./RegisterServlet"* method=*"post"* name=*"validate"* onsubmit="return valid();">

<table style="color: *black*;font-family: *serif*; font-size: *25px*;">

<tr>

<td>User Name</td>

<td><input type=*"text"* name=*"uname"*/></td>

</tr>

<tr>

<td>Password</td>

<td><input type=*"password"* name=*"password"*/></td>

</tr>

<tr>

<td>Confirm Password</td>

<td><input type=*"password"* name=*"cpassword"*/></td>

</tr>

<tr>

<td>Email</td>

<td><input type=*"email"* name=*"email"*/></td>

</tr>

<tr>

<td>Mobile</td>

<td><input type=*"text"* name=*"mobile"* maxlength=*"10"* pattern=*"[789]{1}[0-9]{9}"*/></td>

</tr>

<tr>

<td>Birthday</td>

<td><input type=*"date"* name=*"dob"*/></td>

</tr>

<tr>

<td>Gender</td>

<td><select name=*"gender"*>

<option value=*""*>Select Gender</option>

<option value=*"male"*>Male</option>

<option value=*"female"*>Female</option>

</select>

</td>

</tr>

<tr>

<td>Address</td>

<td><input type=*"text"* name=*"addr"* /></td>

</tr>

<tr>

<td>E-Commerce</td>

<td>

<select name=*"commerce"*>

<option value=*""*>Select E-Commerce</option>

<option value=*"Amazon"*>AmaZon</option>

<option value=*"Flipkart"*>FlipKart</option>

</select>

</td>

</tr>

<tr>

<td colspan=*"2"*><input type=*"submit"* value=*"Submit"*/></td>

</tr>

</table>

</form>

</div>

<ul class=*"actions"*>

</ul>

</div>

</section>

<!-- Scripts -->

<script src=*"assets/js/jquery.min.js"*></script>

<script src=*"assets/js/skel.min.js"*></script>

<script src=*"assets/js/util.js"*></script>

<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->

<script src=*"assets/js/main.js"*></script>

</body>

</html>

**AddProducts.jsp**

<%@page import=*"java.util.ArrayList"*%><!DOCTYPE HTML>

<%@page import=*"sentiment.similarity.analysis.dao.ViewDao"* %>

<%@page import=*"sentiment.similarity.analysis.bean.Bean"* %>

<html>

<head>

<title>AddandViewCatrgory</title>

<meta charset=*"utf-8"* />

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"* />

<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->

<link rel=*"stylesheet"* href=*"assets/css/main.css"* />

<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->

<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->

<script type=*"text/javascript"*>

**function** valid();

{

**var** a= document.forms["product"]["catogery"].value;

**if**(a=="")

{

alert("Please Select Category");

**return** **false**;

}

**var** a= document.forms["product"]["ptitle"].value;

**if**(a=="")

{

alert("Product Title Must be filled");

**return** **false**;

}

**var** a= document.forms["product"]["price"].value;

**if**(a=="")

{

alert("Please Enter Price");

**return** **false**;

}

**var** a= document.forms["product"]["index"].value;

**if**(a=="")

{

alert("Index Must be Filled");

**return** **false**;

}

**var** a= document.forms["product"]["des"].value;

**if**(a=="")

{

alert("Description Must be Filled");

**return** **false**;

}

**var** a= document.forms["product"]["image"].value;

**if**(a=="")

{

alert("Please Select Image");

**return** **false**;

}

}

</script>

</head>

<body class=*"landing"*>

<%ViewDao vi = **new** ViewDao();

ArrayList<Bean> a = vi.sellerViewCategory();

%>

<!-- Header -->

<header id=*"header"* class=*"alt"*>

<h1><a>Sentiment Similarity Analysis</a></h1>

<a href=*"#nav"*>Menu</a>

</header>

<!-- Nav -->

<nav id=*"nav"*>

<ul class=*"links"*>

<li><a href=*"users.jsp"*>View Authorised Users</a></li>

<li><a href=*"FriendRequestAndResponse.jsp"*>Friend Request And Response</a></li>

<li><a href=*"AddandViewCategory.jsp"*>Add and View Category</a></li>

<li><a href=*"AddProducts.jsp"*>Add Products</a></li>

<li><a href=*"ViewAllProductswithRatings.jsp"*>View All Products with Ratings</a></li>

<li><a href=*"ViewAllProductReviews.jsp"*>View All Product Reviews</a></li><li><a href=*"ViewAllStatementSimilarityByReviews.jsp"*>View All Statement Similarity By Reviews</a></li>

<li><a href=*"ViewPurchasedProducts.jsp"*>View Purchased Products</a></li>

<li><a href=*"ViewDeletedFriends.jsp"*>View Deleted Friends</a></li>

<li><a href=*"ViewUserQueryKeyword.jsp"*>View User Query Keyword</a></li>

<li><a href=*"ViewAllProductsConsultByUser.jsp"*>View All Products Consult By User</a></li>

<li><a href=*"ViewAllRecommendedProducts.jsp"*>View All Recommended Products</a></li>

<li><a href=*"ViewAllProductRankResult.jsp"*>View All Product Rank Result</a></li>

<li><a href=*"./LogoutServlet"*>Logout</a></li>

</ul>

</nav>

<!-- Banner -->

<section id=*"banner"*>

<i class=*"icon fa-diamond"*></i>

<h2>Mining Users Trust From E-Commerce Reviews Based On Sentiment Similarity Analysis</h2>

</section>

<!-- One -->

<section id=*"two"* class=*"wrapper special"*>

<div class=*"inner"*>

<header class=*"major narrow"*>

<h2>Add New Category</h2>

<font style="font-family: *serif*;" color=*"black"* size=*"5px;"*>

<%String s= request.getParameter("status");

**if**(s!=**null**){

out.print(s);

}

%>

<form action=*"./SellerAddProductsSer"* method=*"post"* name=*"product"* onsubmit="return valid();">

<table>

<tr>

<td>

Category Name

</td>

<td>

<select name=*"catogery"*>

<option value=*""*>Select Category</option>

<!-- getProductCategory() -->

<%**for**(Bean c:a){ %>

<option value=*"*<%=c.getUname() %>*"*><%=c.getUname() %></option>

<%} %>

</select>

</td>

</tr>

<tr>

<td>

Product Name

</td>

<td>

<input type=*"text"* name=*"ptitle"* required/>

</td>

</tr>

<tr>

<td>

Price

</td>

<td>

<input type=*"text"* name=*"price"* required/>

</td>

</tr>

<tr>

<td>

Index

</td>

<td>

<input type=*"text"* name=*"index"* required/>

</td>

</tr>

<tr>

<td>

Description

</td>

<td>

<input type=*"text"* name=*"des"* required/>

</td>

</tr>

<tr>

<td>

Select Image

</td>

<td>

<input type=*"file"* name=*"image"* required/>

</td>

</tr>

<tr>

<td colspan=*"2"*>

<input type=*"submit"* value=*"Submit"*>

</td>

</tr>

</table>

</form>

</font>

</header>

<div class=*"image-grid"*>

</div>

<ul class=*"actions"*>

</ul>

</div>

</section>

<!-- Scripts -->

<script src=*"assets/js/jquery.min.js"*></script>

<script src=*"assets/js/skel.min.js"*></script>

<script src=*"assets/js/util.js"*></script>

<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->

<script src=*"assets/js/main.js"*></script>

</body>

</html>

**ViewAllRecommendedProducts.jsp**

<%@page import=*"java.util.ArrayList"*%><!DOCTYPE HTML>

<%@page import=*"sentiment.similarity.analysis.dao.ViewDao"* %>

<%@page import=*"sentiment.similarity.analysis.bean.Bean"* %>

<html>

<head>

<title>AddandViewCatrgory</title>

<meta charset=*"utf-8"* />

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1"* />

<!--[if lte IE 8]><script src="assets/js/ie/html5shiv.js"></script><![endif]-->

<link rel=*"stylesheet"* href=*"assets/css/main.css"* />

<!--[if lte IE 8]><link rel="stylesheet" href="assets/css/ie8.css" /><![endif]-->

<!--[if lte IE 9]><link rel="stylesheet" href="assets/css/ie9.css" /><![endif]-->

<script type=*"text/javascript"*>

**function** valid()

{

**var** x = document.forms["recomended"]["recomended"].value;

**if**(x=="")

{

alert("Please Select Search Keyword");

**return** **false**;

}

}

**function** fun1(val)

{

//alert("val")

obj=**new** XMLHttpRequest();

obj.open("post","./ViewRecomendedProductsServlet?id="+val,**true**)

obj.send()

obj.onreadystatechange=fun2

}

**function** fun2()

{

**if**(obj.readyState==4)

{

//alert("obj.responseText")

document.getElementById('recomended').innerHTML=(obj.responseText)

}

}

</script>

</head>

<body class=*"landing"*>

<%ViewDao vi = **new** ViewDao();

ArrayList<Bean> al = vi.sellerViewCategory();

ArrayList<Bean> al1 = (ArrayList) session.getAttribute("al");

**int** i = 0;

%>

<!-- Header -->

<header id=*"header"* class=*"alt"*>

<h1><a>Sentiment Similarity Analysis</a></h1>

<a href=*"#nav"*>Menu</a>

</header>

<!-- Nav -->

<nav id=*"nav"*>

<ul class=*"links"*>

<li><a href=*"users.jsp"*>View Authorised Users</a></li>

<li><a href=*"FriendRequestAndResponse.jsp"*>Friend Request And Response</a></li>

<li><a href=*"AddandViewCategory.jsp"*>Add and View Category</a></li>

<li><a href=*"AddProducts.jsp"*>Add Products</a></li>

<li><a href=*"ViewAllProductswithRatings.jsp"*>View All Products with Ratings</a></li>

<li><a href=*"ViewAllProductReviews.jsp"*>View All Product Reviews</a></li>

<li><a href=*"ViewAllStatementSimilarityByReviews.jsp"*>View All Statement Similarity By Reviews</a></li>

<li><a href=*"ViewPurchasedProducts.jsp"*>View Purchased Products</a></li>

<li><a href=*"ViewDeletedFriends.jsp"*>View Deleted Friends</a></li>

<li><a href=*"ViewUserQueryKeyword.jsp"*>View User Query Keyword</a></li>

<li><a href=*"ViewAllProductsConsultByUser.jsp"*>View All Products Consult By User</a></li>

<li><a href=*"ViewAllRecommendedProducts.jsp"*>View All Recommended Products</a></li>

<li><a href=*"ViewAllProductRankResult.jsp"*>View All Product Rank Result</a></li>

<li><a href=*"./LogoutServlet"*>Logout</a></li>

</ul>

</nav>

<!-- Banner -->

<section id=*"banner"*>

<i class=*"icon fa-diamond"*></i>

<h2>Mining Users Trust From E-Commerce Reviews Based On Sentiment Similarity Analysis</h2>

</section>

<!-- One -->

<section id=*"two"* class=*"wrapper special"*>

<div class=*"inner"*>

<header class=*"major narrow"*>

<font size=*"3px;"* style="font-family: *serif*;" color=*"black"*>

<h3>View All Recommended Products</h3>

<form action=*"./ViewRecomendedProductsServlet"* name=*"recomended"* method=*"post"* onsubmit="return valid();">

<table>

<tr>

<td>Select Category:</td>

<td><!--<select name="recomended" id="recomended" onchange="fun1(this.value)" required>

-->

<select name=*"recomended"*>

<option value=*""*>Select Category</option>

<%**for**(Bean b:al){ %>

<option value=*"*<%=b.getProductCategory()%>*"*><%=b.getProductCategory()%></option>

<%} %>

</select>

</td>

<td><input type=*"submit"* value=*"Submit"*/></td>

</tr>

</table>

</form>

</font>

</header>

<div class=*"image-grid"*>

</div>

<ul class=*"actions"*>

</ul>

</div>

</section>

<!-- One -->

<section id=*"two"* class=*"wrapper special"*>

<div class=*"inner"* style="margin-top: *-200px*;">

<header class=*"major narrow"*>

<font style="font-family: *serif*;" color=*"black"*>

<h3>Your Result</h3>

<table>

<tr>

<th>S.No</th>

<th>Product Name</th>

<th>Recommended By</th>

<th>Recommended To</th>

<th>Date Of Recommended</th>

</tr>

<%**for**(Bean b1 :al1){ %>

<tr>

<td><%= ++i%></td>

<td><%=b1.getProductTitle() %></td>

<td><%=b1.getUname() %></td>

<td><%=b1.getUtype() %></td>

<td><%=b1.getDob()%></td>

</tr>

<%} %>

</table>

</font>

</form>

</header>

<div class=*"image-grid"*>

</div>

<ul class=*"actions"*>

</ul>

</div>

</section>

<!-- Footer -->

<footer id=*"footer"*>

<div class=*"inner"*>

<ul class=*"copyright"*>

</ul>

</div>

</footer>

<!-- Scripts -->

<script src=*"assets/js/jquery.min.js"*></script>

<script src=*"assets/js/skel.min.js"*></script>

<script src=*"assets/js/util.js"*></script>

<!--[if lte IE 8]><script src="assets/js/ie/respond.min.js"></script><![endif]-->

<script src=*"assets/js/main.js"*></script>

</body>

</html>

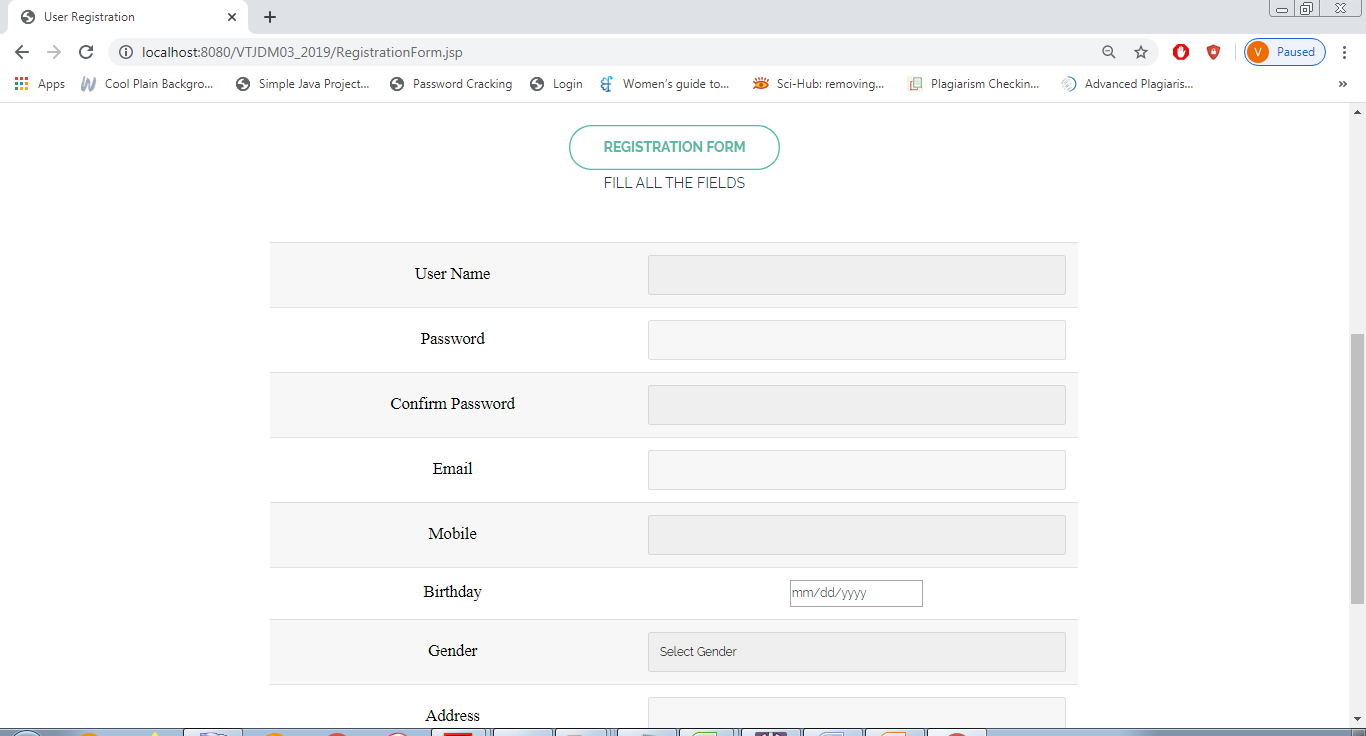
**CHAPTER 7**

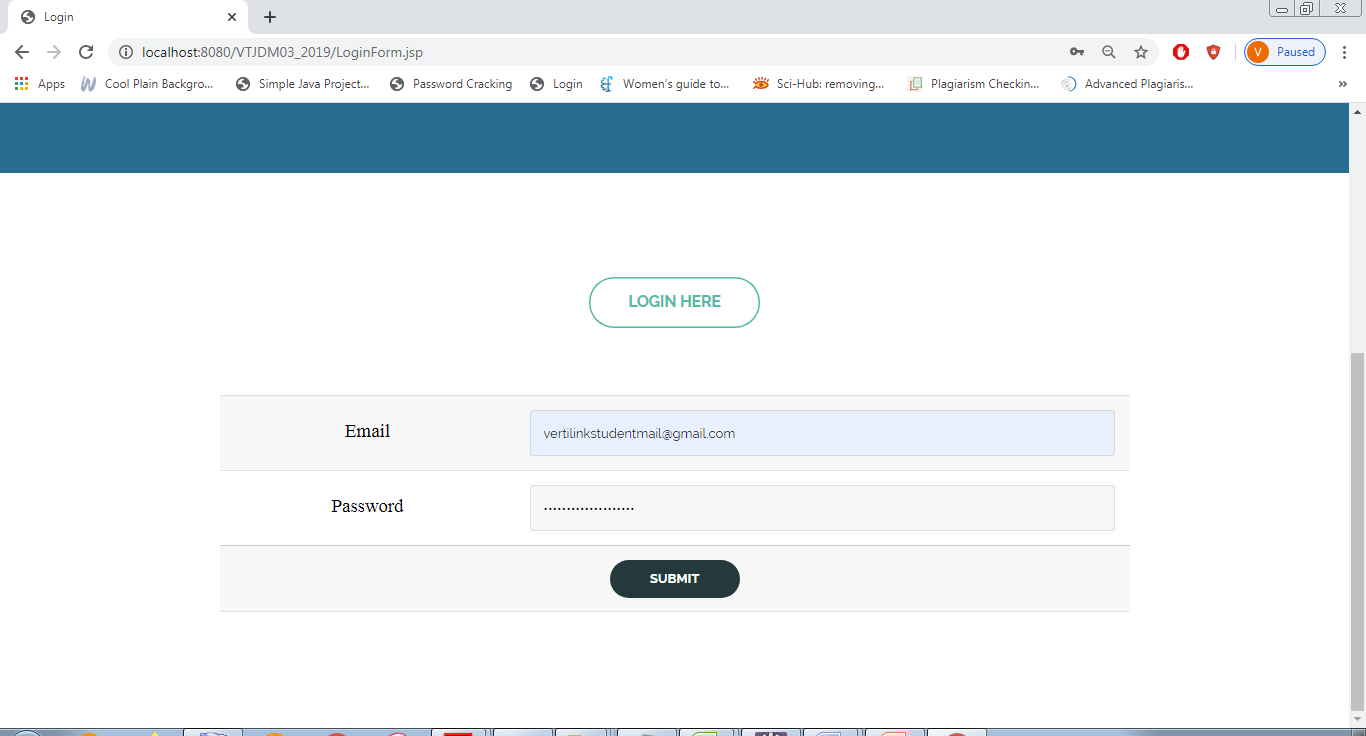
**SNAPSHOTS**

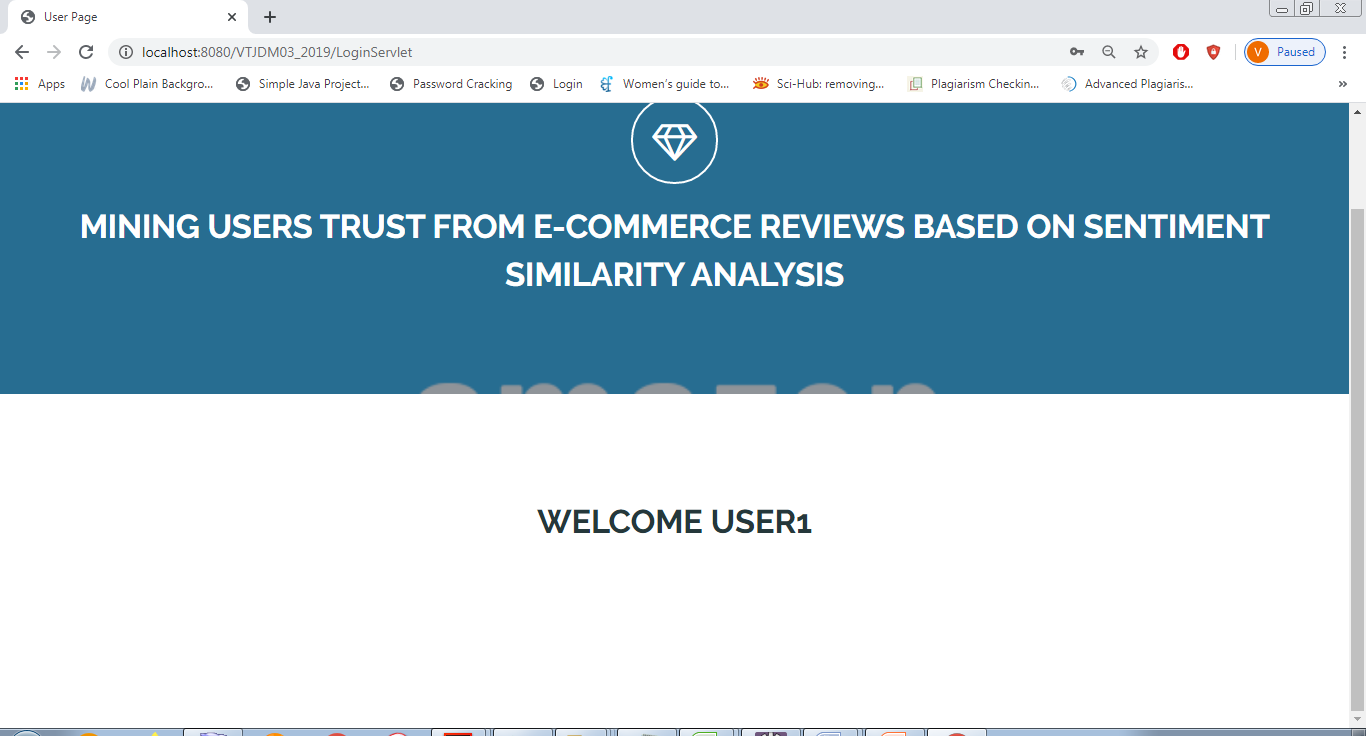
**General:**

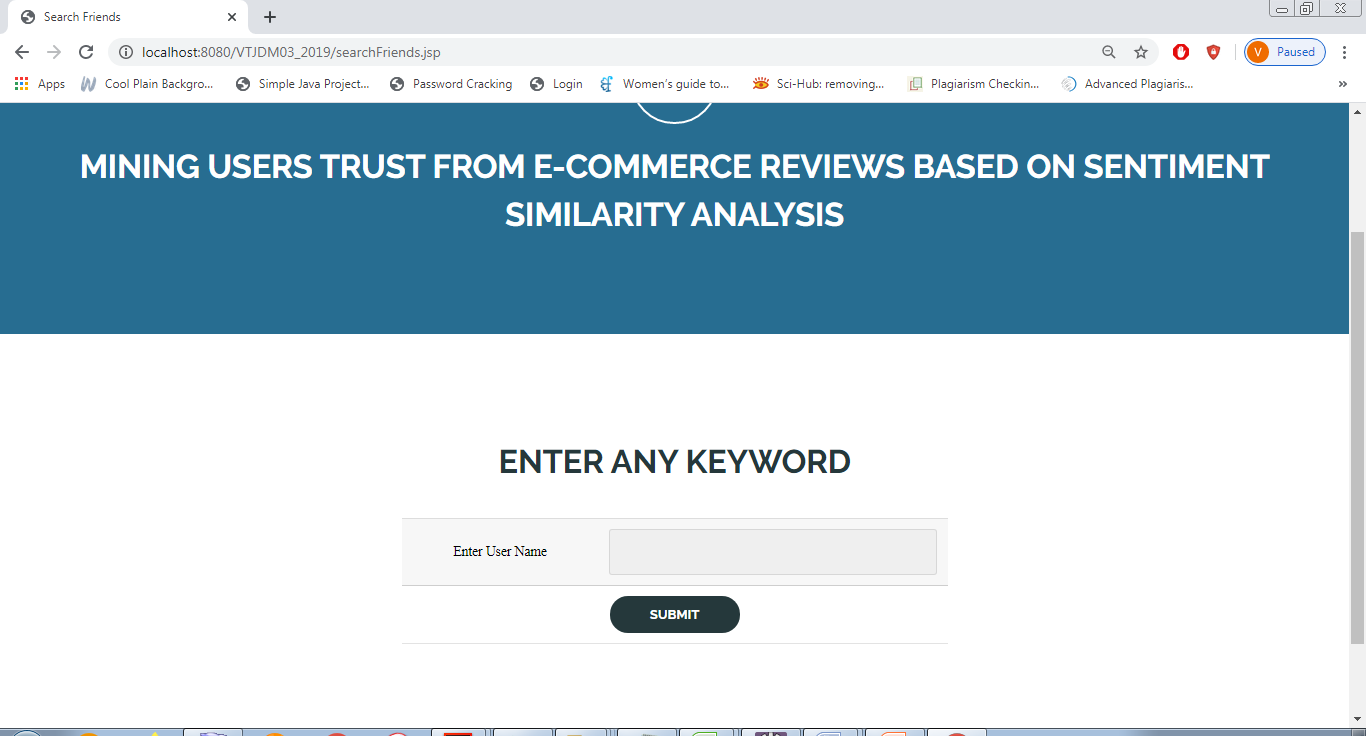
This project is implements like web application using COREJAVA and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

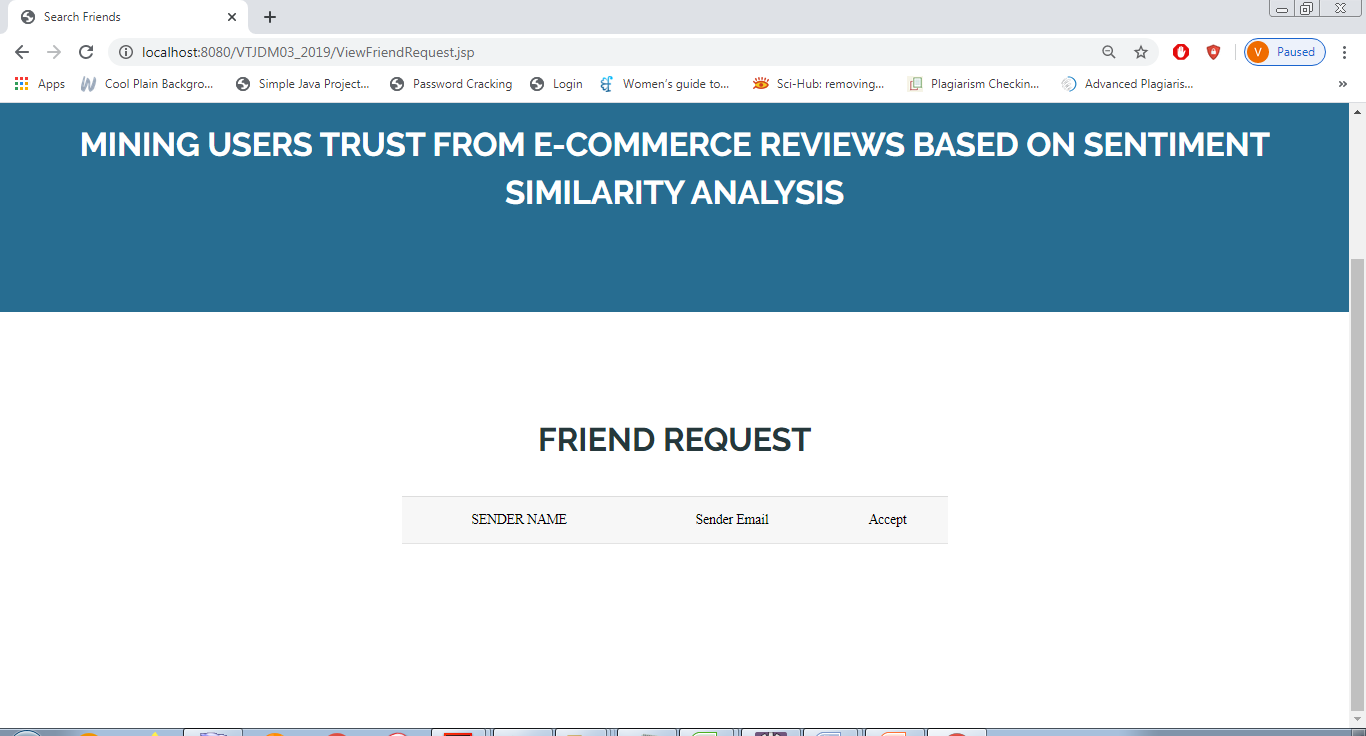
**SNAPSHOTS**

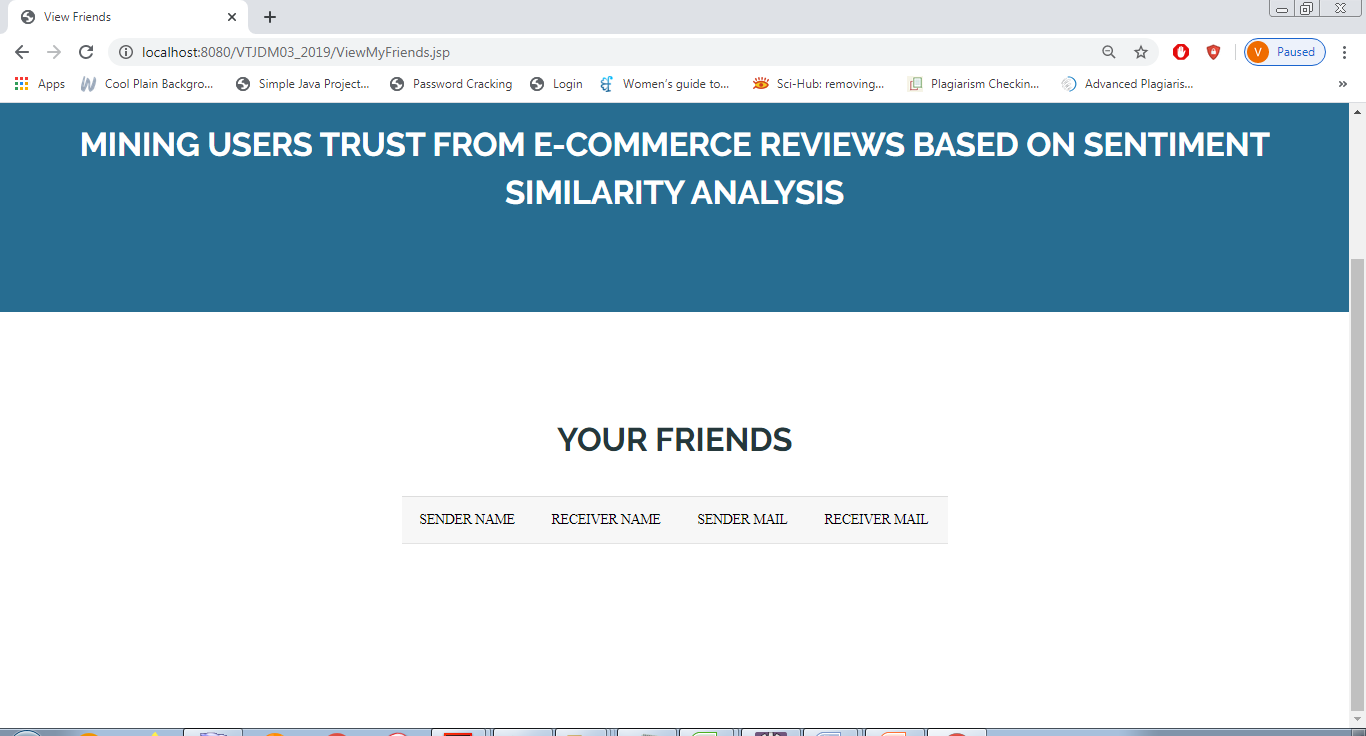


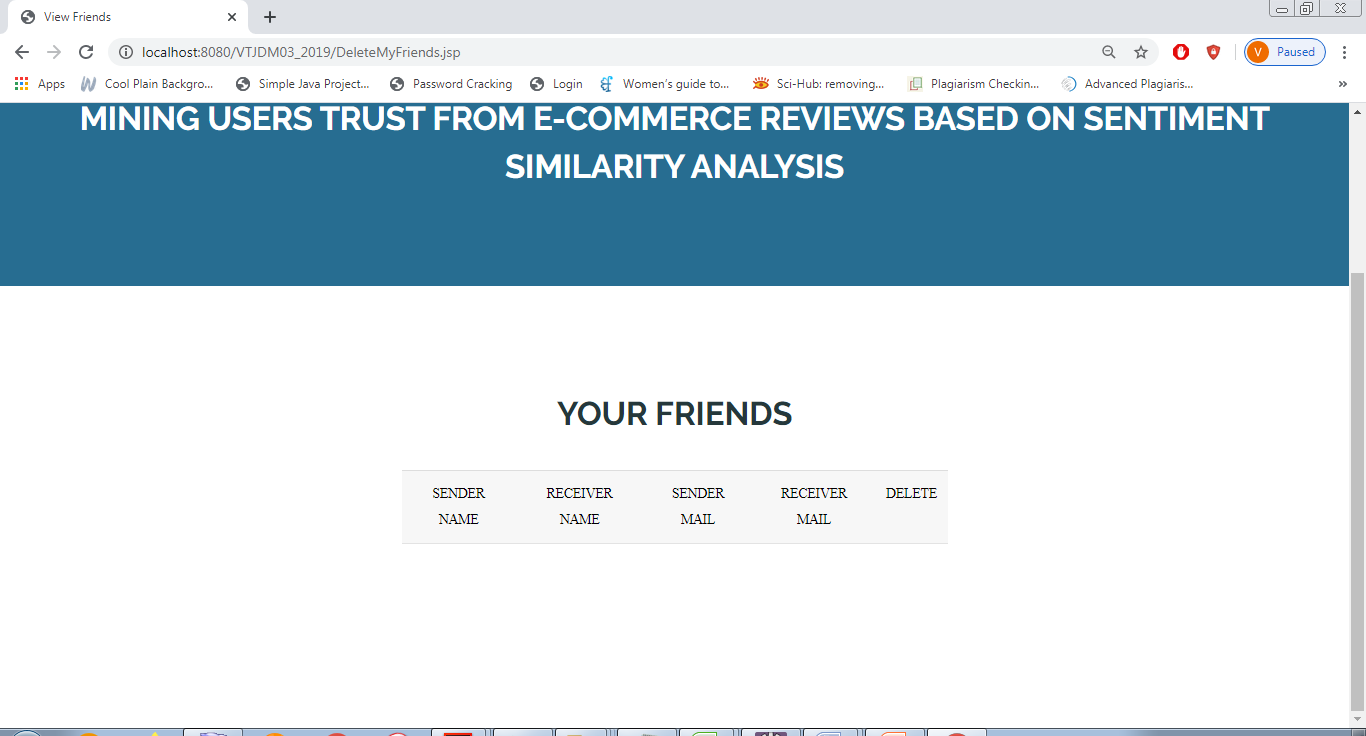


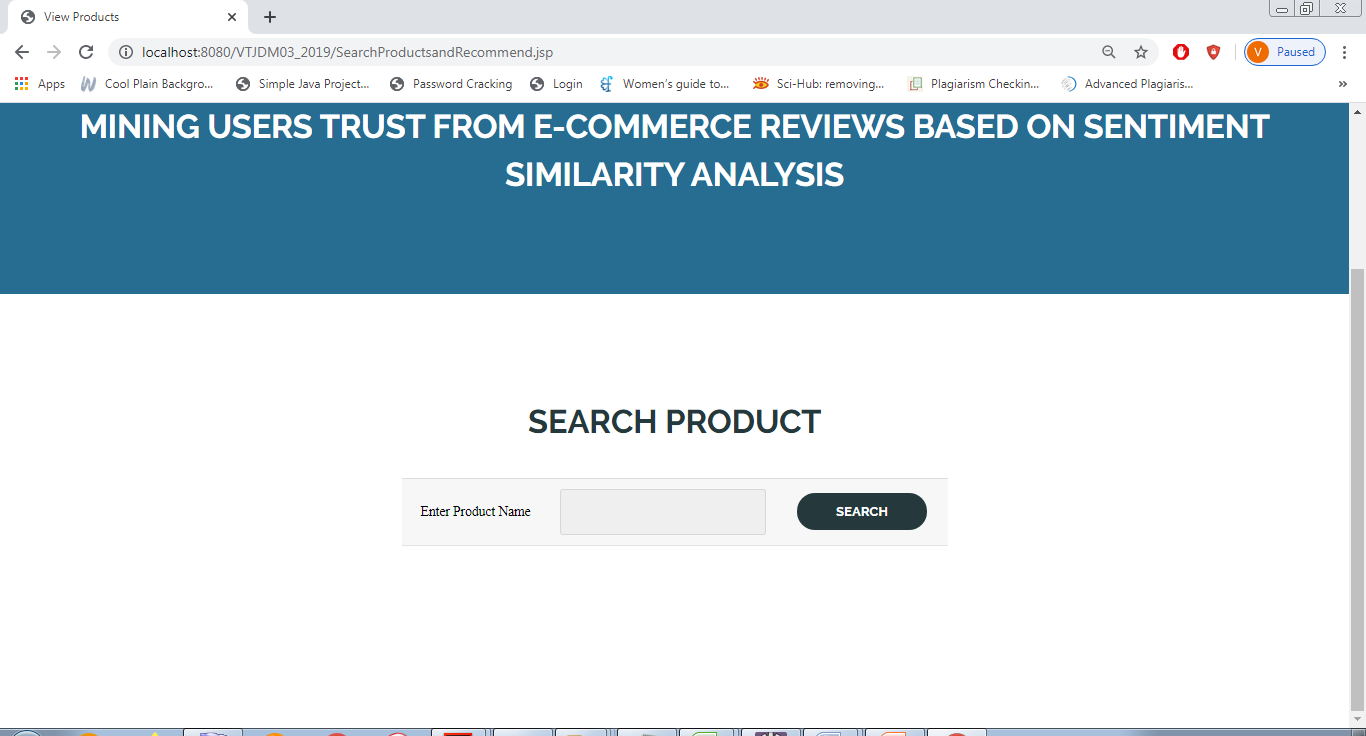


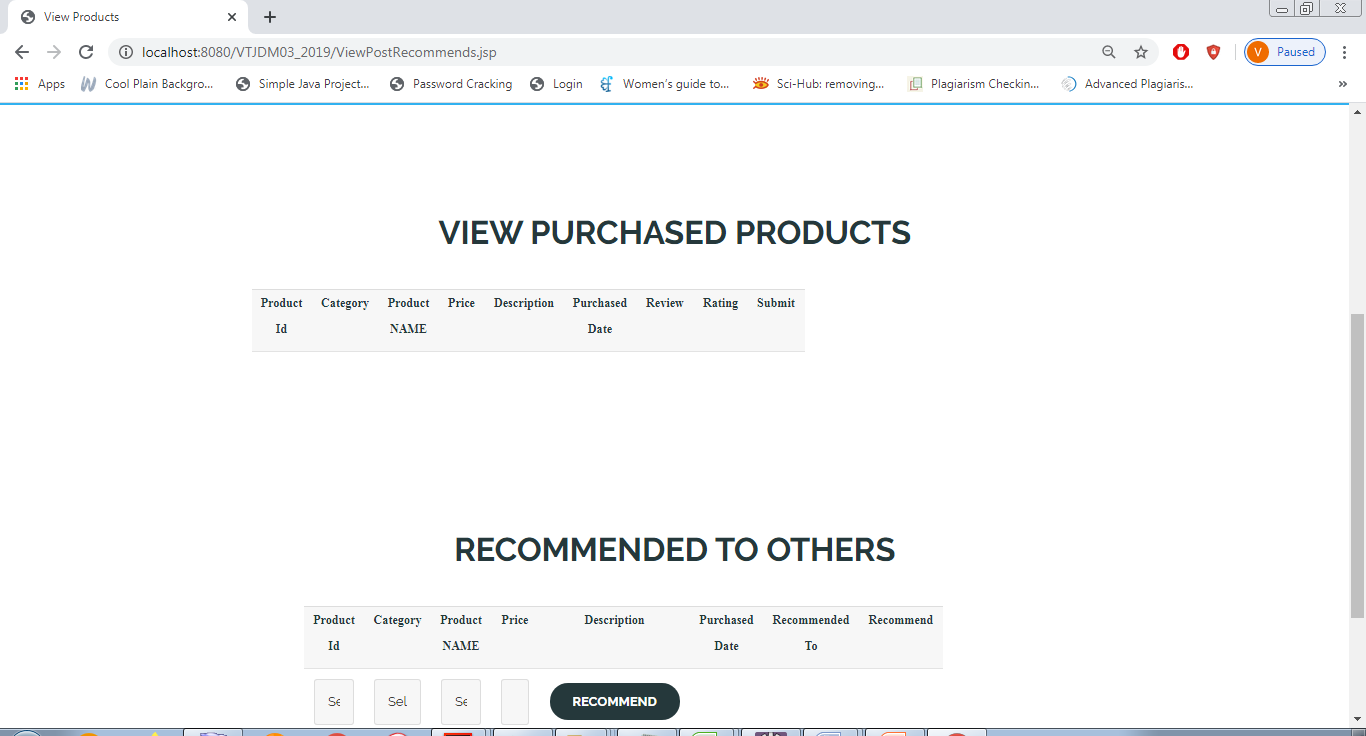


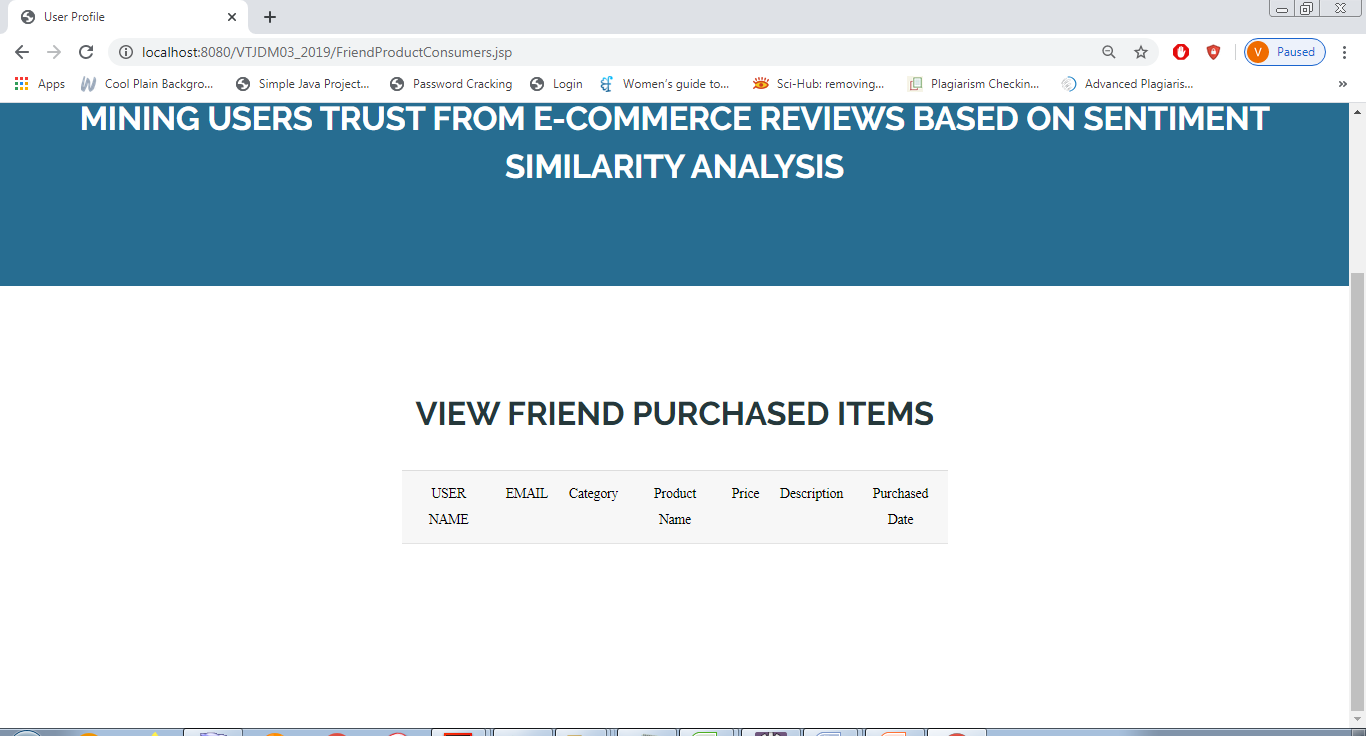


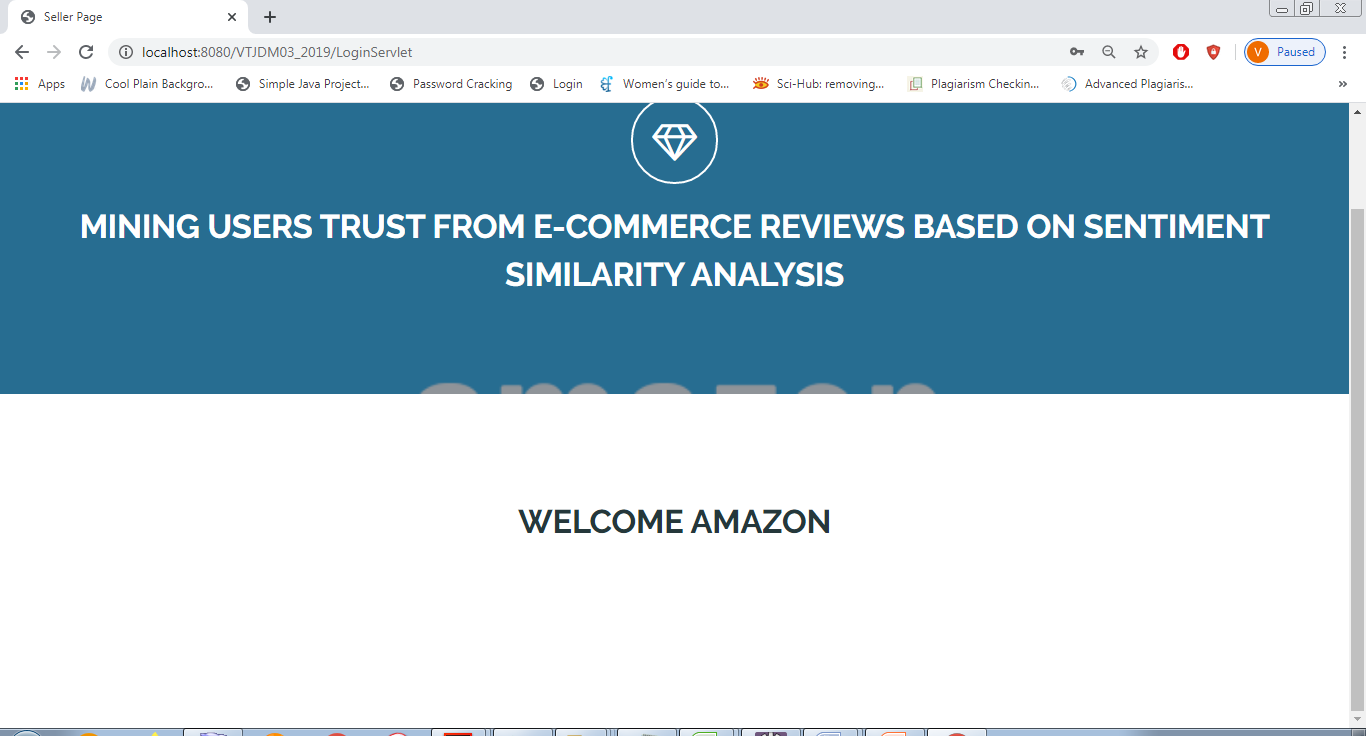


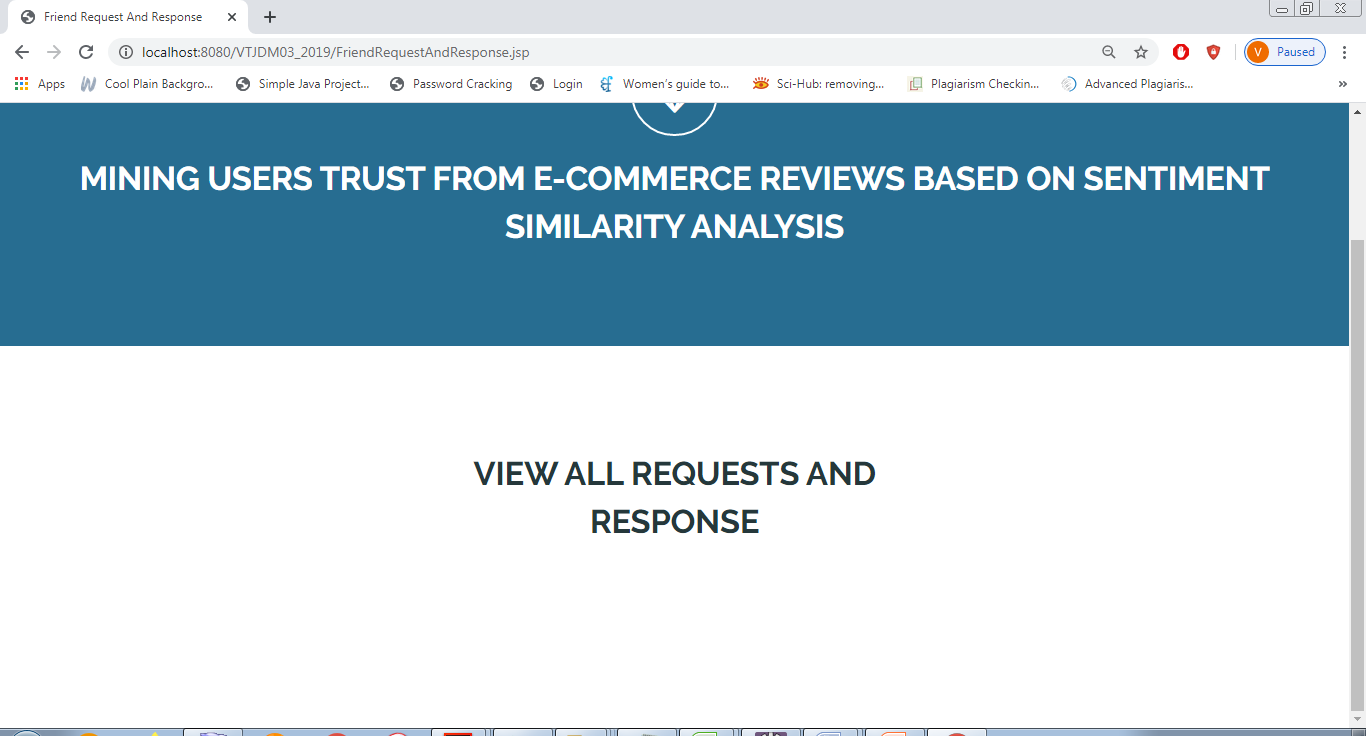


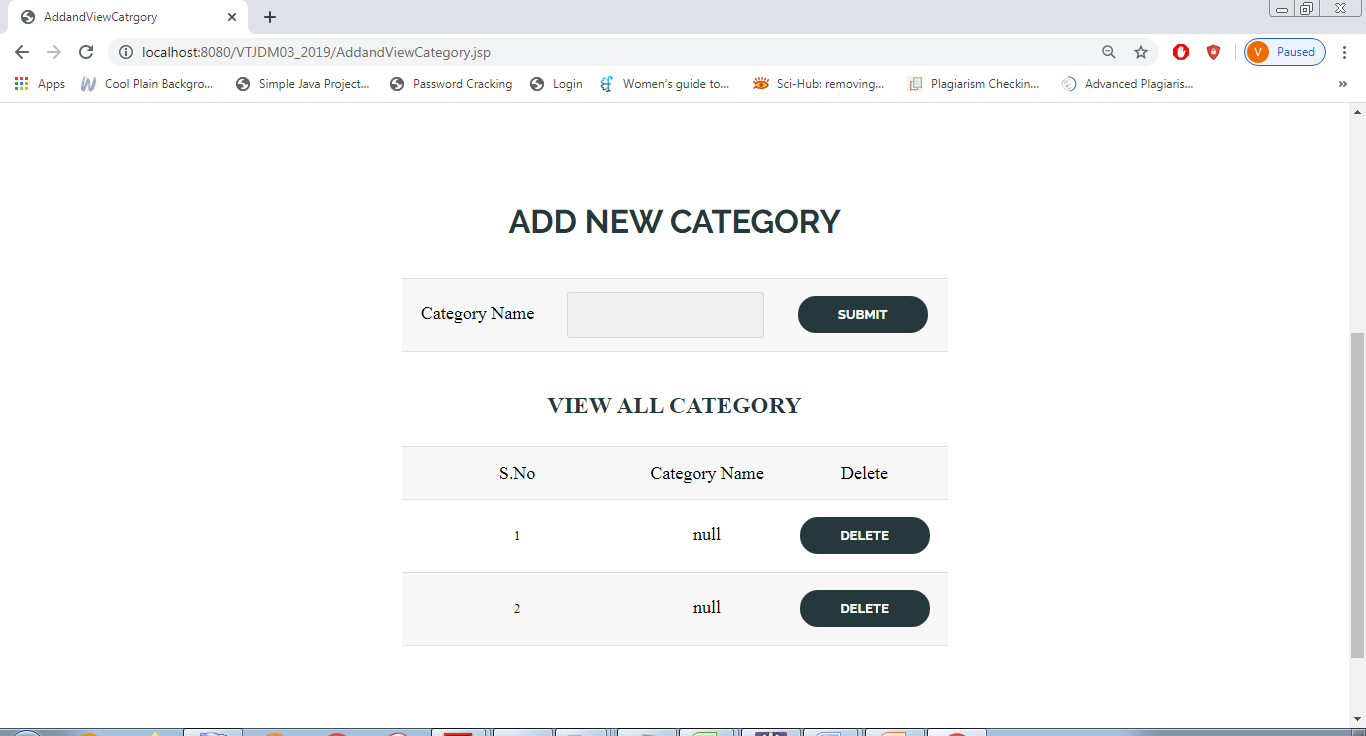


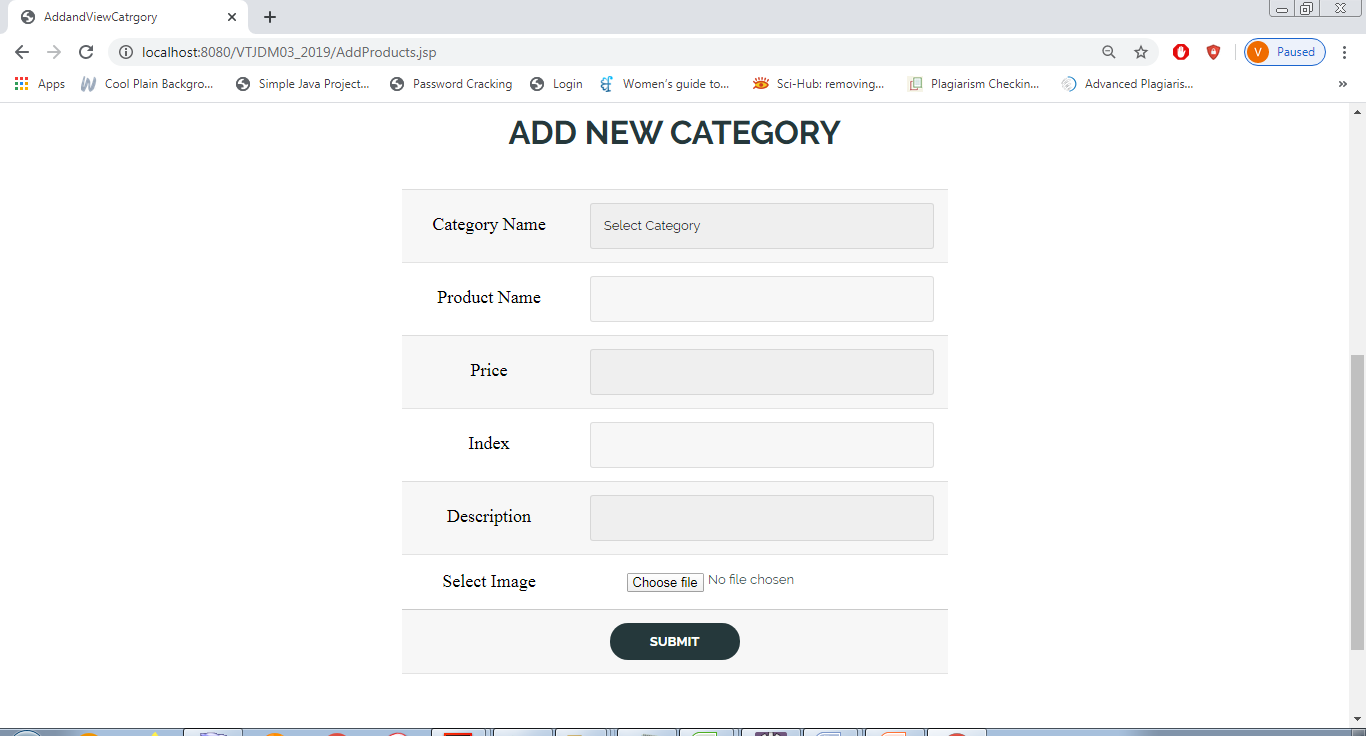


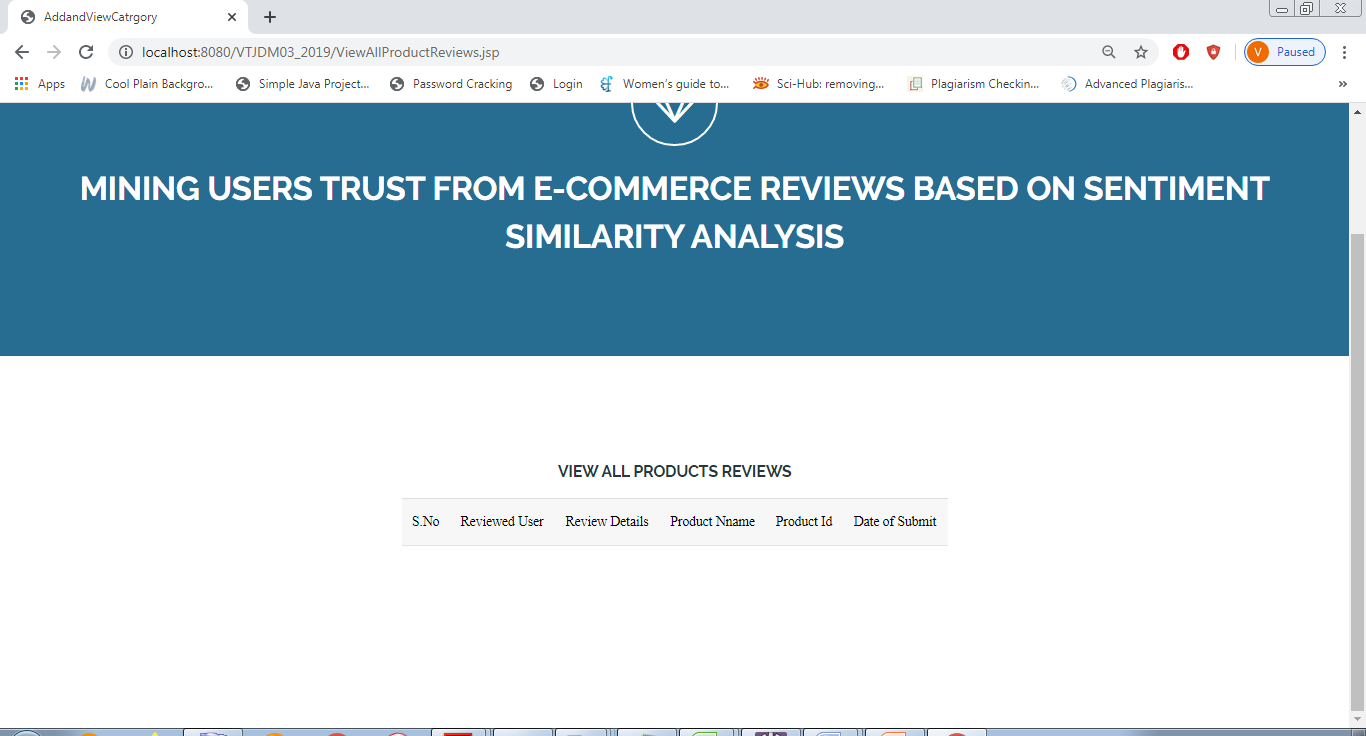


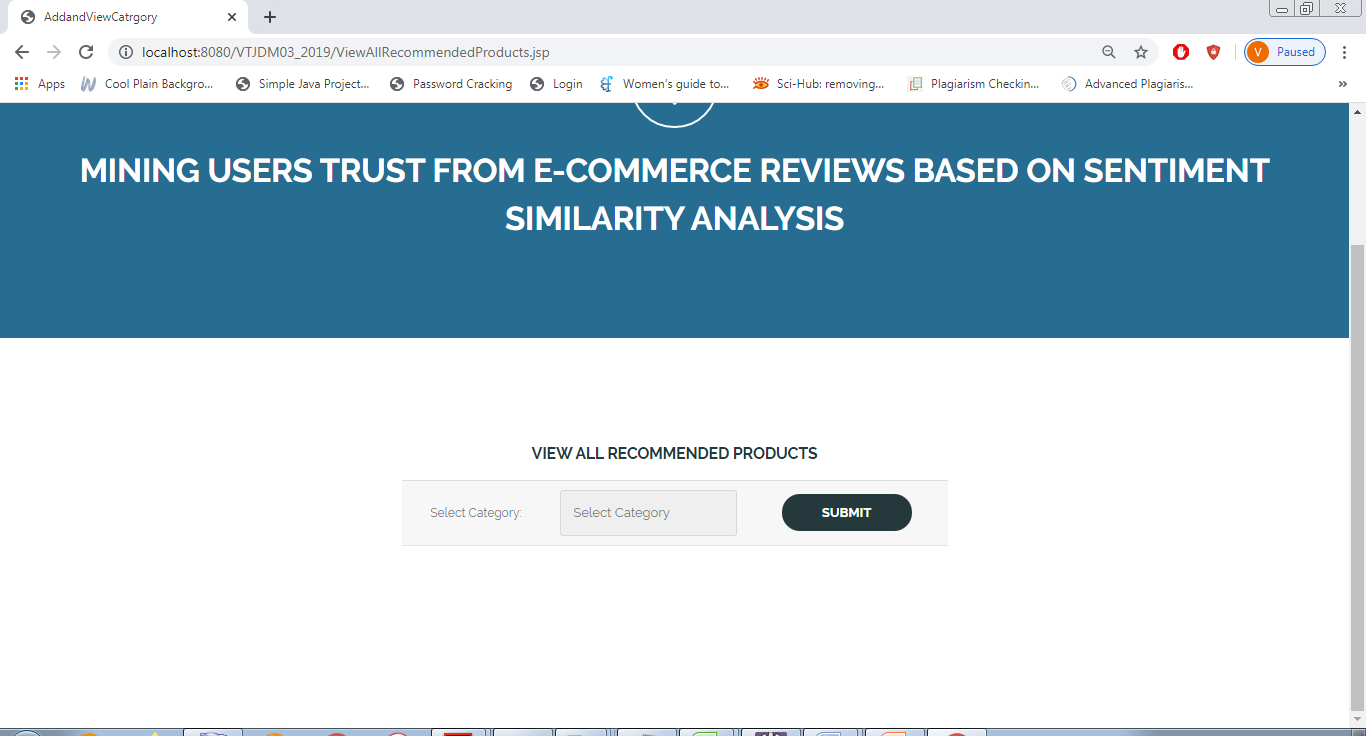


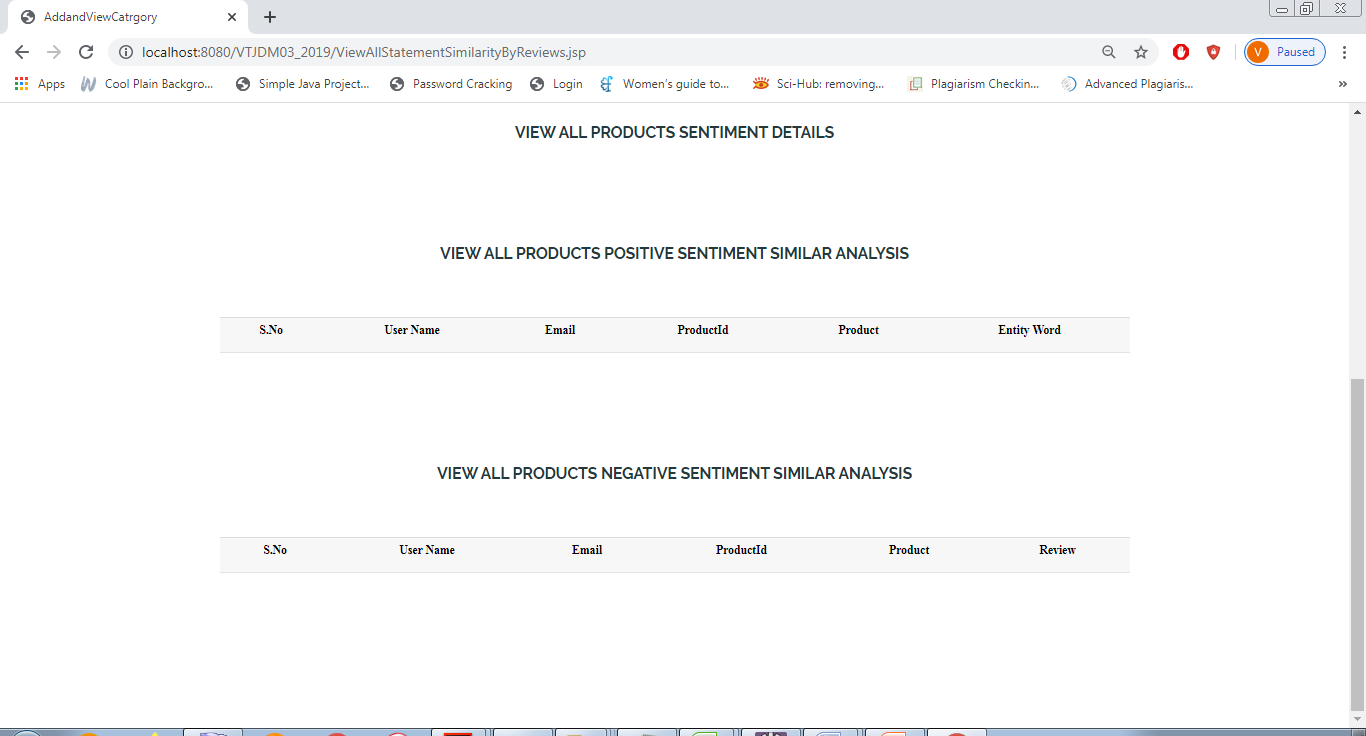












**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

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

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3 Types of Tests**

**8.3.1 Unit testing**

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

**8.3.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

**8.3.3 System Test**

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

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

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

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**CHAPTER 9**

**APPLICATION**

**9.1 GENERAL**

The main goal of the existing sentiment analysis methods is to cluster the sentiments of users, commonly dividing people's sentiments to things into several types. Even at the entity and feature levels, its main purpose is to divide the user's sentiments into likes or dislikes. However, the above methods concern directly in overall trend which is insufficient when we calculate the trust based on sentiment similarity. It is necessary to analyze the specific attitudes on specific objects in reviews.

**9.2** **FUTURE ENHANCEMENT**

It can be said with certainty that the user's trust relationship can be obtained through the similarity of them. But the user's trust is not simple normal linear with the user's similarity. How to accurately describe this relationship will be the focus in further research. At the same time, there are several valuable study of sentiment similarity and trust in E-commerce field in the future: (1) Not each user gives their reviews on each item, so the user's reviews data are usually sparse for a particular item. how to explore similarity of users with extremely sparse reviews data, e.g. by designing more efficient algorithm to overcome the challenge; (2) The degree to which people trust others is different for different things. Under more stringent requirements, it is also necessary to distinguish the categories of trust targets in details. how to include other information, for example, purchase item category, brand and other activities, into user sentiment calculation framework and (3) how to incorporate temporal factors to capture users' similarity change will be the focus of future research.

**CHAPTER 10**

**CONCLUSION & REFERENCE**

**10.1 CONCLUSION**

In our work, we address the problem of mining users trust in E-commerce system. By defining two kinds of trust relationship, namely, direct trust and propagation trust, we transfer the point of exploring trust between users into calculation of sentiment similarity of their reviews. With the help of entity-sentiment word pairs mining, sentiment similarity of reviews can be calculated and direct trust relationships can be obtained through sentiment similarity analysis, which contains of sentiments and ratings aspect. These two aspects can be used jointly to analyse the sentiment direct trust relationship. We establish a weighed trust graph model for propagation trust computing. Propagation trust is the use of the propagation characteristics of trust. It is an indirect trust between two users without direct trust and is obtained through intermediate users who have direct trust between these two users. The propagation trust calculation approach is based on the improved shortest path algorithm, and the time complexity is O(V2), where V is the number of node in the graph. Ways to improve the computational complexity of the algorithm is a new problem that needs further study because the relatively large number of users in modern e-commerce system.

**10.2 REFERENCE:**

[1] H. Liu, F. Xia, Z. Chen, N. Y. Asabere, J. Ma, and R. Huang, ``TruCom: Exploiting domain-specific trust networks for multicategory item recommendation,'' IEEE Syst. J., vol. 11, no. 1, pp. 295304, Mar. 2017.

[2] P.-Y. Hsu, H.-T. Lei, S.-H. Huang, T. H. Liao, Y.-C. Lo, and C.-C. Lo, ``Effects of sentiment on recommendations in social network,'' in Electron Markets. Berlin, Germany: Springer, 2018, pp. 110, doi: 10.1007/s12525- 018-0314-5.

[3] C. Qin, W. Siyi, and A. Lin, ``The joint beta distribution with refund rate in online C2C trust building: A theoretical study on Taobao,'' in Proc. Int. Conf. E-Learn. E-Technol. Educ. (ICEEE), Lodz, Poland, Sep. 2012, pp. 191196.

[4] S. Kraounakis, I. N. Demetropoulos, A. Michalas, M. S. Obaidat, P. G. Sarigiannidis, and M. D. Louta, ``A robust reputation-based computational model for trust establishment in pervasive systems,'' IEEE Syst. J., vol. 9, no. 3, pp. 878891, Sep. 2015.

[5] P. De Meo, E. Ferrara, D. Rosaci, and G. M. L. Sarné, ``Trust and compactness in social network groups,'' IEEE Trans. Cybern., vol. 45, no. 2, pp. 205216, Feb. 2015.

[6] M. G. Ozsoy and F. Polat, ``Trust based recommendation systems,'' in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM), Niagara Falls, ON, Canada, Aug. 2013, pp. 12671274.

[7] L. Sheugh and S. H. Alizadeh, ``A fuzzy aproach for determination trust threshold in recommender systems based on social network,'' in Proc. 9th Int. Conf. E-Commerce Developing Countries, Focus E-Bus. (ECDC), Isfahan, Iran, Apr. 2015, pp. 15.

[8] Y. Ruan, L. Alfantoukh, and A. Durresi, ``Exploring stock market using twitter trust network,'' in Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl., Gwangiu, South Korea, Mar. 2015, pp. 428433.

[9] C.-N. Ziegler, Social Web Artifacts for Boosting Recommenders, vol. 487. Berlin, Germany: Springer, 2013.

[10] C.-N. Ziegler and J. Golbeck, ``Investigating interactions of trust and interest similarity,'' Decis. Support Syst., vol. 43, no. 2, pp. 460475, 2007.

[11] N. Lathia, S. Hailes, and L. Capra, ``Trust-based collaborative ltering,'' in Trust Management II, vol. 263, Y. Karabulut, J. Mitchell, P. Herrmann, and C. D. Jensen, Eds. Boston, MA, USA: Springer, 2008, pp. 119134.

[12] D. H. Alahmadi and X.-J. Zeng, ``ISTS: Implicit social trust and sentiment based approach to recommender systems,'' Expert Syst. Appl., vol. 42, no. 22, pp. 88408849, 2015.

[13] M. Koufaris and W. Hampton-Sosa, ``The development of initial trust in an online company by new customers,'' Inf. Manage., vol. 41, no. 3, pp. 377397, 2004.

[14] H. Zhong, S. Zhang, Y. Wang, and Y. Shu, ``Study on directed trust graph based recommendation for E-commerce system,'' Int. J. Comput. Commun. Control, vol. 9, no. 4, pp. 510523, 2014.

[15] M. Li, Y. Xiang, B. Zhang, and Z. Huang, ``A sentiment delivering estimate scheme based on trust chain in mobile social network,'' Mobile Inf. Syst., vol. 2015, Sep. 2015, Art. no. 745095, doi: 10.1155/2015/745095.

[16] A. Jøsang, R. Ismail, and C. Boyd, ``A survey of trust and reputation systems for online service provision,'' Decision Support Syst., vol. 43, no. 2, pp. 618644, 2007.

[17] B. Zhang, R. Yong, M. Li, J. Pan, and J. Huang, ``A hybrid trust evaluation framework for E-commerce in online social network: A factor enrichment perspective,'' IEEE Access, vol. 5, pp. 70807096, 2017, doi: 10.1109/ACCESS.2017.2692249.

[18] D. Alahmadi and X.-J. Zeng, ``Improving recommendation using trust and sentiment inference from OSNs,'' Int. J. Knowl. Eng., vol. 1, no. 1, pp. 917, 2015.

[19] Y.-J. Li and Y.-F. Dai, ``Research on trust mechanism for peer-to-peer network,'' Chin. J. Comput., vol. 33, no. 3, pp. 390405, 2010.

[20] B. Zhang, Y. Xiang, and Q. Xu, ``Semantics based information trust computation and propagation algorithm for semantic Web,'' in Proc. 5th Int. Conf. Wireless Commun., Netw. Mobile Comput., Beijing, China, Sep. 2009, pp. 14.

[21] Y. Sun, Z. Han, and K. J. R. Liu, ``Defense of trust management vulnerabilities in distributed networks,'' IEEE Commun. Mag., vol. 46, no. 2, pp. 112119, Feb. 2008.

[22] P. H. Andersen and R. Kumar, ``Emotions, trust and relationship development in business relationships: A conceptual model for buyerseller dyads,'' Ind. Marketing Manage., vol. 35, no. 4, pp. 522535, 2006.

[23] T. DeWitt, D. T. Nguyen, and R. Marshall, ``Exploring customer loyalty following service recovery: The mediating effects of trust and emotions,'' J. Service Res., vol. 10, no. 3, pp. 269281, 2008.

[24] F. Alam and A. Paul, ``A computational model for trust and reputation relationship in social network,'' in Proc. Int. Conf. Recent Trends Inf. Technol. (ICRTIT), Chennai, India, Apr. 2016, pp. 16.

[25] S. Li, I. Doh, and K. Chae, ``Non-redundant indirect trust search algorithm based on a cross-domain trust model in content delivery network,'' in Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT), Bongpyeong, South Korea, Feb. 2017, pp. 7277.

[26] F. Yuan, H. Li, and Q. Fu, ``A new trust propagation and aggregation method for P2P networks,'' in Proc. 2nd Int. Conf. Instrum., Meas., Com- put., Commun. Control, Harbin, China, Dec. 2012, pp. 11761181.

[27] B. Liu, ``Sentiment analysis and opinion mining,'' Synthesis Lectures Hum. Lang. Technol., vol. 5, no. 1, pp. 1167, 2012.

[28] S. Y. Ganeshbhai and B. K. Shah, ``Feature based opinion mining: A survey,'' in Proc. IEEE Int. Advance Comput. Conf. (IACC), Banglore, India, Jun. 2015, pp. 919923.

[29] R. Ren, D. D. Wu, and T. Liu, ``Forecasting stock market movement direction using sentiment analysis and support vector machine,'' IEEE Syst. J., to be published, doi: 10.1109/JSYST.2018.2794462.

[30] J. Kikuchi and V. Klyuev, ``Gathering user reviews for an opinion dictionary,'' in Proc. 18th Int. Conf. Adv. Commun. Technol. (ICACT), Pyeongchang, South Korea, Jan./Feb. 2016, pp. 566569.

[31] T. Takehara, S. Miki, N. Nitta, and N. Babaguchi, ``Extracting context information from microblog based on analysis of online reviews,'' in Proc. IEEE Int. Conf. Multimedia Expo Workshops, Melbourne, VIC, Australia, Jul. 2012, pp. 248253.

[32] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, ``Sentiment analysis and classication based on textual reviews,'' in Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES), Chennai, India, Feb. 2013, pp. 271276.

[33] J. Akaichi, ``Social networks' Facebook' statutes updates mining for sentiment classication,'' in Proc. Int. Conf. Social Comput., Alexandria, VA, USA, Sep. 2013, pp. 886891.

[34] F. Colace, M. De Santo, and L. Greco, ``Sentiment mining through mixed graph of terms,'' in Proc. 17th Int. Conf. Netw.-Based Inf. Syst., Salerno, Italy, Sep. 2014, pp. 324330.

[35] H.-Y. Kao and Z.-Y. Lin, ``A categorized sentiment analysis of Chinese reviews by mining dependency in product features and opinions from blogs,'' in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Tech- nol., Toronto, ON, Canada, Aug./Sep. 2010, pp. 456459.

[36] L. Yang, Y. Zhang, C. Xing, and T. Zhang, `A node interest similarity based P2P trust model,'' in Proc. IEEE 12th Int. Conf. Commun. Technol., Nanjing, China, Nov. 2010, pp. 572575.

[37] H. Zhang, Y. Wang, and X. Zhang, ``Transaction similarity-based contextual trust evaluation in E-commerce and E-service environments,'' in Proc. IEEE Int. Conf. Web Services, Washington, DC, USA, Jul. 2011, pp. 500507.

[38] G. Wang, X. Gui, and G. Wei, ``A recommendation trust model based on E-commerce transactions content-similarity,'' in Proc. Int. Conf. Mach. Vis. Hum.-Mach. Interface, Kaifeng, China, Apr. 2010, pp. 105108.

[39] J. Golbeck, ``Trust and nuanced prole similarity in online social networks,'' ACM Trans. Web, vol. 3, no. 4, pp. 130, 2009.

[40] H. Mohammadhassanzadeh and H. R. Shahriari, ``Using user similarity to infer trust values in social networks regardless of direct ratings,'' in Proc. 9th Int. Conf. Inf. Secur. Cryptol. (ISC), Tabriz, Iran, Sep. 2012, pp. 6672.

[41] M. R. Garakani and M. Jalali, ``A trust prediction approach by using collaborative ltering and computing similarity in social networks,'' in Proc. Int. Congr. Technol., Commun. Knowl. (ICTCK), Mashhad, Iran, Nov. 2014, pp. 14.

[42] Q. Zhang, Y. Zhuo, and Z. Gong, ``A trust inspection model based on society behavior similarity rule in dynamic networks,'' in Proc. Int. Conf. Comput. Sci. Softw. Eng., Hubei, China, Dec. 2008, pp. 970973.

[43] G. Cai, M. Wang, Y. Wang, and Y. Zhou, ``A collusion detection trust model based on behaviour similarity,'' in Proc. IET Int. Conf. Inf. Commun. Technol. (IETICT), Beijing, China, Apr. 2013, pp. 241245.

[44] K. Khan and B. B. Baharudin, ``Analysis of syntactic patterns for identification of features from unstructured reviews,'' in Proc. 4th Int. Conf. Intell. Adv. Syst. (ICIAS), Kuala Lumpur, Malaysia, Jun. 2012, pp. 165169.

[45] Z. Zhai, H. Xu, J. Li, and P. Jia, ``Sentiment classification for Chinese reviews based on key substring features,'' in Proc. Int. Conf. Natural Lang. Process. Knowl. Eng., Dalian, China, Sep. 2009, pp. 18.

[46] K. Bloom and S. Argamon, ``Automated learning of appraisal extraction patterns,'' Lang. Comput., vol. 71, no. 1, pp. 249260, 2009.

[47] S. Robertson, ``Understanding inverse document frequency: On theoretical arguments for IDF,'' J. Document., vol. 60, no. 5, pp. 503520, 2004.

[48] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, ``Image-based recommendations on styles and substitutes,'' in Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR), Santiago, Chile, Aug. 2015, pp. 4352.

[49] J. McAuley, R. Pandey, and J. Leskovec, ``Inferring networks of substitutable and complementary products,'' in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Sydney, NSW, Australia, Aug. 2015, pp. 785794.

[50] L. Havrlant and V. Kreinovich, ``A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation),'' Int. J. General Syst., vol. 46, no. 1, pp. 2736, 2017.