

**ACHIEVING DATA
TRUTHFULNESS AND PRIVACY
PRESERVATION IN DATA
MARKETS**

Submitted in partial fulfilment of the
requirements for the award of Bachelor of
Science degree in Computer Science

By Doney I G lawren (Reg. No.39290022)
Chandan Prakash (Reg. No.39290015)



**DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI - 600 119
MARCH – 2022**



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Doney I G lawren (Reg. No.39290022) Chandan Prakash (Reg. No.39290015)** who carried out the project entitled "**Achieving data truthfulness and privacy preservation in data markets**" under my supervision from _____ to _____.

Internal Guide - Dr. K. Mohana Prasad

Head of the Department - Dr. L. Lakshmanan M.E., Ph.D., and Dr.S. Vigneshwari M.E., Ph.D.

Submitted for Viva voce Examination held

on _____

Internal Examiner

External Examiner 2

DECLARATION

I, Doney I G lawren (Reg. No.39290022) and Chandan Prakash (Reg. No.39290015) hereby declare that the Project Report entitled " Achieving data truthfulness and privacy preservation in data markets" done by me under the guidance of Dr. K. Mohana Prasad is submitted in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. K. Mohana Prasad**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways For the Completion

SYNOPSIS

ABSTRACT:

Data mining applications have obtained massive growth in today's internet era. It eventually leads to a serious threat to the security of an individual's personal and sensitive information. Raw data of millions of users are collected by the service providers through data contributors and they are shared with the data consumers. Existing systems use Privacy Preserving data mining (PPDM) which modifies the data without compromising the security of the sensitive information contained in the data. In this paper, we have proposed a new technique called Truthfulness Preserving Data Mining (TPDM) which is structured as encryptthen-sign fashion. It uses partial homomorphism encryption and identity-based signature that provides batch-verification and data confidentiality. Performance of the model was evaluated using Yahoo! Music rating dataset and 2009 RECS dataset. Performance measures show that TPDM incurs low computation and communication overheads when compared to existing models. Keywords Natural hazards, Disaster Response, Drones, Clustering, Flood Monitoring, classification

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE
1	INTRODUCTION	1
1.1	About the Project	1
2	SYSTEM ANALYSIS	2
2.1	Existing system	2
2.2	Proposed system	2
3	REQUIREMENTS SPECIFICATION	3
3.1	Introduction	4
3.2	Hardware and Software specification	5
3.3	Technologies Used	5
3.3.1	Java	6
3.3.3.1	Introduction to Java	6
4	Project Purpose and Scope	6
4.1	Purpose	7
4.2	Project Scope	7
4.3	Product Perspective	7
4.4	System Features	8
4.5	Design and Implementation Constraints	9

4.6	Other Nonfunctional Requirements	10
5	SYSTEM DESIGN	12
5.1	Architecture Diagram	13
6	SYSTEM DESIGN – DETAILED	19
6.1	Modules	20
6.2	Module explanation	20
7	CODING AND TESTING	21
7.1	Coding	22
7.2	Coding standards	25
7.3	Test procedure	26
7.4	Test data and output	29
	SOURCE CODE	30
	SNAP SHOTS	35
	REFERENCES	44

1.INTRODUCTION

Aim:

The main aim of this project to provide effective survey that guarantees data truthfulness and privacy preservation.

Synopsis:

In the era of big data, society has developed an insatiable appetite for sharing personal data. Realizing the potential of personal data's economic value in decision making and user experience enhancement, several open information platforms have emerged to enable personspecific data to be exchanged on the Internet.

To integrate truthfulness and privacy preservation in a practical data market, there are four major challenges. The first and the thorniest design challenge is that verifying the truthfulness of data collection and preserving the privacy seem to be contradictory objectives. Ensuring the truthfulness of data collection allows the data consumers to verify the validities of data contributors' identities and the content of raw data, whereas privacy preservation tends to prevent them from learning these confidential contents.

Yet, another challenge comes from data processing, which makes verifying the truthfulness of data collection even harder. Nowadays, more and more data markets provide data services rather than directly offering raw data.

The third challenge lies in how to guarantee the truthfulness of data processing, under the information asymmetry between the data consumer and the service provider due to data confidentiality. Last but not least, the fourth design challenge is the efficiency requirement of data markets, especially for data

acquisition, i.e., the service provider should be able to collect data from a large number of data contributors with low latency.

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

In the existing system some service data providers don't do survey the user's data properly, some providers will change the data for their financial greed and some providers will submit the data to consumers without processing it to reach their target.

2.2 PROPOSED SYSTEM

In this project, we have proposed the first efficient secure scheme TPDM for data markets, which simultaneously guarantees data truthfulness and privacy preservation. In TPDM, the data contributors have to truthfully submit their own data, but cannot impersonate others. Besides, the service provider is enforced to truthfully collect and process data. Furthermore, both the personally identifiable information and the sensitive raw data of data contributors are well protected. In addition, we have instantiated TPDM with two different data services, and extensively evaluated their performances on two real-world datasets.

3.REQUIREMENT SPECIFICATIONS

3.1 INTRODUCTION

The growth of the internet has made millions of people to access numerous online applications eventually leading to sharing their personal and sensitive data with the service providers. Service providers gain full authority of using this personal information shared using the online applications such as the images, videos and the documents that the users send to one another. These data are then issued to any third party, commonly known as data consumers, who use these sensitive data for their purposes. Data markets have become very popular which is the acts as a hub where the data providers sell the user information and the data consumers buy this critical information.

One of the major challenges of these data markets is privacy preservation. Numerous technologies have evolved but the truthfulness of the data is not yet guaranteed. When it comes to a service provider, the main of him would be to maximize his profit with low computational expenditure. To attain this some synthetic data are added to the raw datasets and given to the data consumers. They even return a fake dataset without processing the raw data obtained to them. If these acts are not properly identified and prohibited then serious threats could occur to the data consumers and in turn to the data markets.

To eliminate such misuse of sensitive data, truthfulness and privacy preservation of data plays a vital role in data markets. By ensuring the truthfulness of the collected data, the data consumers can validate and verify their data. There are four main challenges that are evolved around maintaining and preserving the data in a

data market. The first and the foremost is to ensure the truthfulness of the collected data. The second challenge comes while processing the collected data. The third challenge lies in guarantying data confidentiality between data consumer and data provider.

3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.2.1 HARDWARE REQUIREMENTS

- Hard Disk : 200GB and Above
- RAM : 2GB and Above
- Processor : i3 and Above

3.2.2 SOFTWARE REQUIREMENTS

- Windows 7 and Above
- JDK 1.7
- Tomcat 7.0
- MySQL 5.0

3.3 TECHNOLOGIES USED

- J2EE (JSP, Servlets)
- JavaScript, HTML, CSS, AJAX
- Web Services (JAX -WS)

3.3.1 JAVA

- Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office),

was intended to replace C++, although the feature set better resembles that of Objective C.

3.3.1.1 INTRODUCTION TO JAVA

- Java has been around since 1991, developed by a small team of Sun

Microsystems developers in a project originally called the green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

- The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialled in order to answer any ringing telephone from any other phone in the office.
- Around the time the First-Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by NC SA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users

to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

- It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

4.1 Project Purpose and Scope

4.1 Purpose

TPDM is designed to simultaneously guarantee data truthfulness and privacy preservation of the raw data obtained from the data contributors. In TPDM, data contributors register their data with the registration centers and verify that their submitted data is truthful. The data contributors cannot impersonate other data contributor's data. The service provider is enforced to truthfully collect the data and process it for further verification. Both the identities of the data contributors and the sensitive information about the data are well protected. The proposed model is extensively evaluated through two real-world datasets and their performance is evaluated.

4.2 Project Scope

To the best of our knowledge, TPDM is the first secure mechanism for data markets achieving both data truthfulness and privacy preservation. _ TPDM is structured internally in a way of Encrypt then- Sign using partially homomorphic encryption and identity-based signature. It enforces the service provider to truthfully collect and to process real data. Besides, TPDM incorporates a two-layer batch verification scheme with an efficient outcome verification scheme, which can drastically reduce computation overhead. _ We instructively instantiate TPDM with two kinds of practical data services, namely profile matching and data distribution.

This paper proposes a TPDM model that is used to secure the data markets. It simultaneously guarantees data truthfulness along with privacy preservation. In this proposed model the all the data contributors. in TPDM, all the data contributors are made to submit their own data truthfully. The service providers are also made to collect and process the data securely. The performance of the model is evaluated by using two real-world datasets. The evaluation results have shown the efficiency of the model when large datasets are used. Future works may include the use of various machine learning techniques to make the model more feasible to complex datasets and handle the growing privacy threats

4.3 Product Perspective

One of the major challenges of these data markets is privacy preservation. Numerous technologies have evolved but the truthfulness of the data is not yet guaranteed. When it comes to a service provider, the main of him would be to maximize his profit with low computational expenditure. To attain this some synthetic data are added to the raw datasets and given to the data consumers. They

even return a fake dataset without processing the raw data obtained to them. If these acts are not properly identified and prohibited then serious threats could occur to the data consumers and in turn to the data markets.

To eliminate such misuse of sensitive data, truthfulness and privacy preservation of data plays a vital role in data markets. By ensuring the truthfulness of the collected data, the data consumers can validate and verify their data. There are four main challenges that are evolved around maintaining and preserving the data in a data market. The first and the foremost is to ensure the truthfulness of the collected data. The second challenge comes while processing the collected data. The third challenge lies in guarantying data confidentiality between data consumer and data provider. Here the service provider develops symmetric/asymmetric cryptosystem that can be used by the data consumers to encrypt their raw data. The final challenge is concerned with the efficiency requirement of the data markets.

Existing models use PPDM (Privacy preservation data mining) that is intended to modify the raw data without compromising the security of the sensitive information [2]. In this paper, we have proposed a technique called TPDM (Truthfulness and Privacy Preservation in Data Markets). TPDM take advantage of using partial homomorphic encryption which is further used to construct a ciphertext space. This facilitates the service providers to initiate their data services and then allows the data consumers to validate and verify the correctness of data processing results without compromising the confidentiality of the data. To the best of our knowledge TPDM serves as the first technique to preserve both privacy and data confidentiality in data markets.

4.4 System Features

We consider a two-layer system model for data markets. The model has a data acquisition layer and a data trading layer. There are four major kinds of entities, including data contributors, a service provider, data consumers, and a registration center. In the data acquisition layer, the service provider procures massive raw data from the data contributors, such as social network users, mobile smart devices, smart meters, and so on. In order to incentivize more data contributors to actively submit high-quality data, the service provider needs to reward those valid ones to compensate their data collection costs.

The registration centre maintains an online database of registrations, and assigns each registered data contributor an identity and a password. Besides, she maintains an official website, called certificated bulletin board, on which the legitimate system participants can publish essential information, e.g., white lists, blacklists, resubmit-lists, and reward-lists of data contributors. Yet, another duty of the registration centre is to set up the parameters for a signature scheme and a cryptosystem. To avoid being a single point of failure or bottleneck, redundant registration centers, which have identical functionalities and databases, can be installed.

4.5 Design and Implementation Constraints

4.5.1 Constraints in Analysis

- ☐ Constraints as Informal Text
- ☐ Constraints as Operational Restrictions
- ☐ Constraints Integrated in Existing Model Concepts
- ☐ Constraints as a Separate Concept
- ☐ Constraints Implied by the Model Structure

4.5.2 Constraints in Design

- ☐ Determination of the Involved Classes

- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

4.5.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore, it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many objectoriented methods.

4.6 Other Non-functional Requirements

4.6.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

4.6.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

5. SYSTEM DESIGN

5.1 Architecture Diagram:

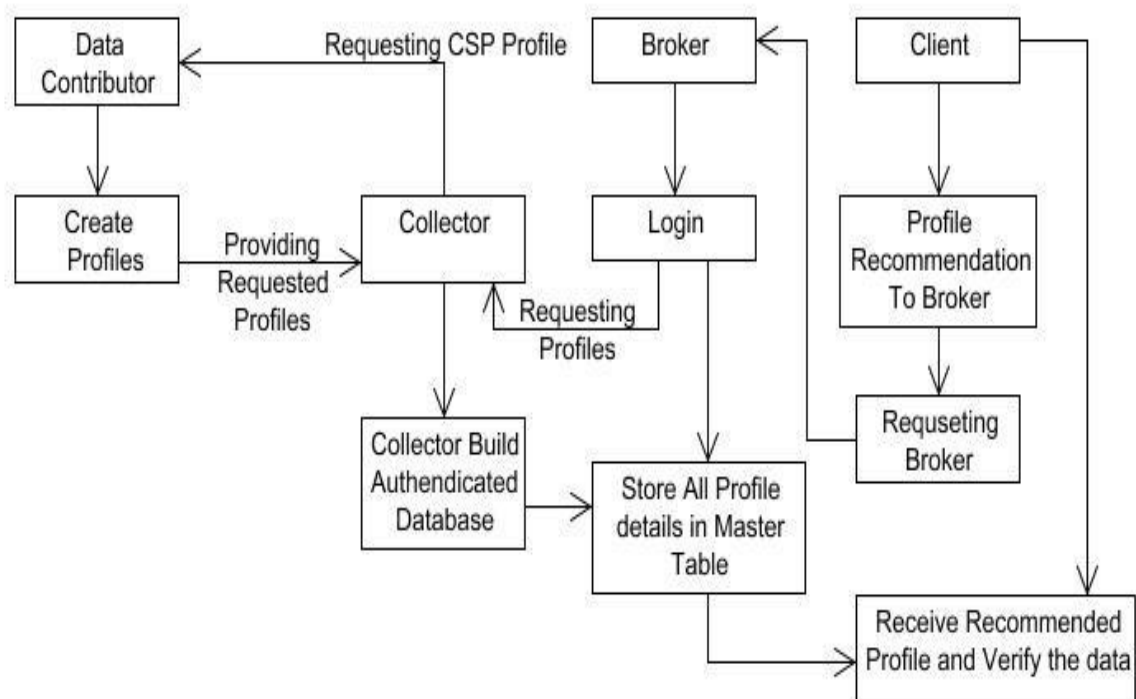
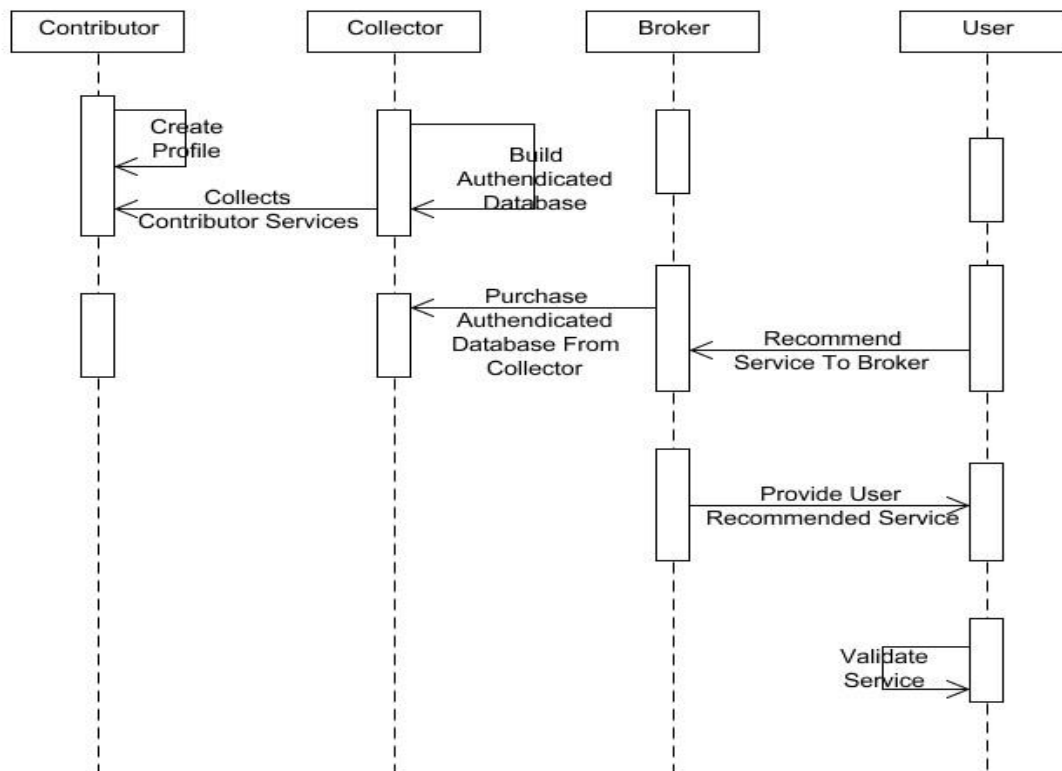


Fig: 5.1

5.2 Sequence Diagram:\

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



5.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

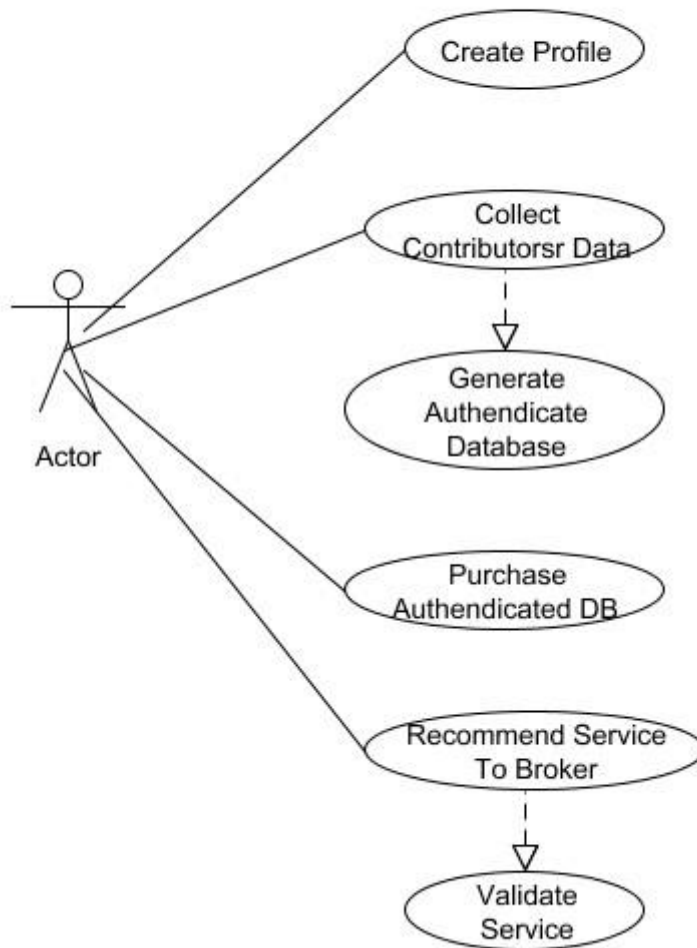
5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

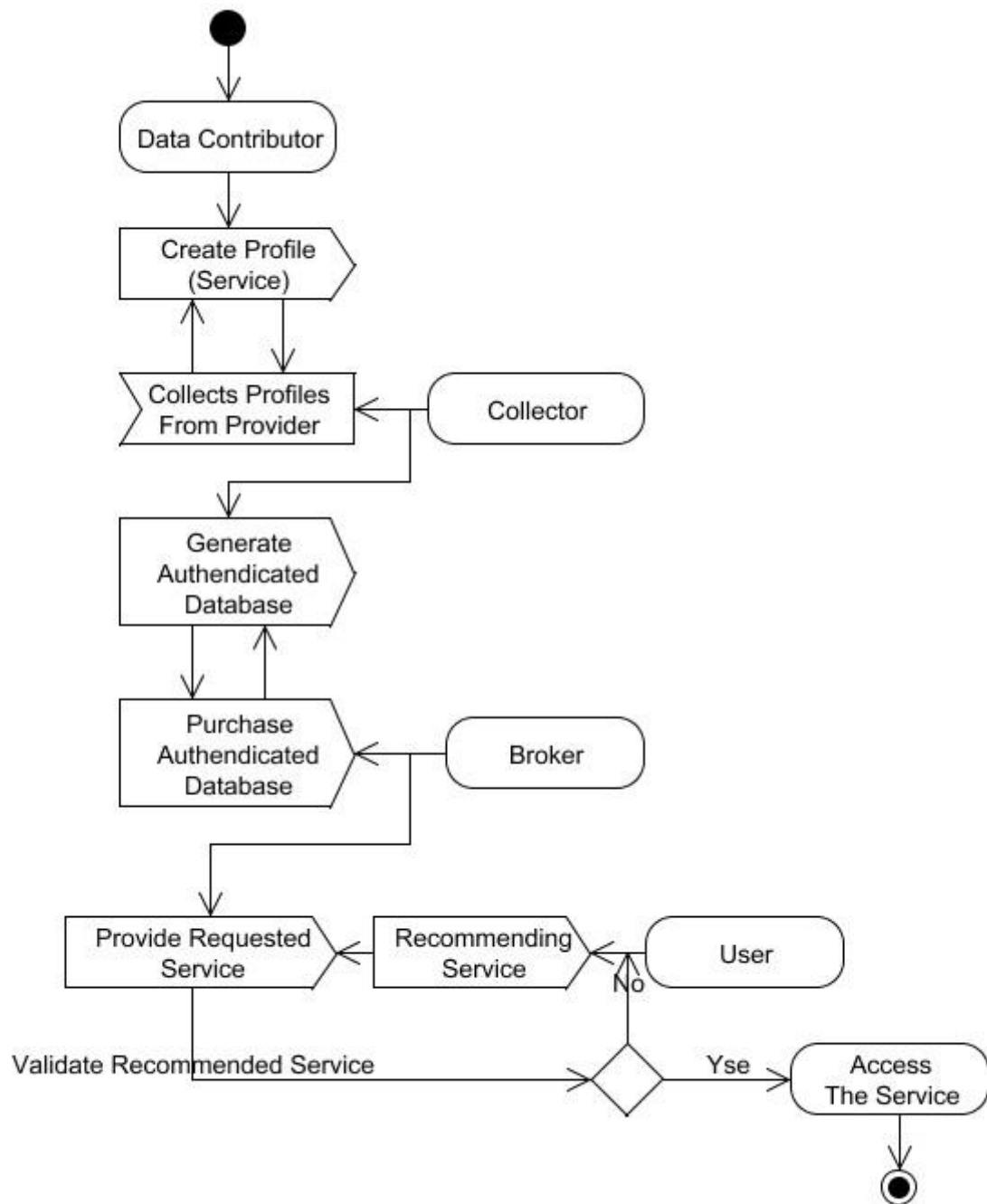


5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

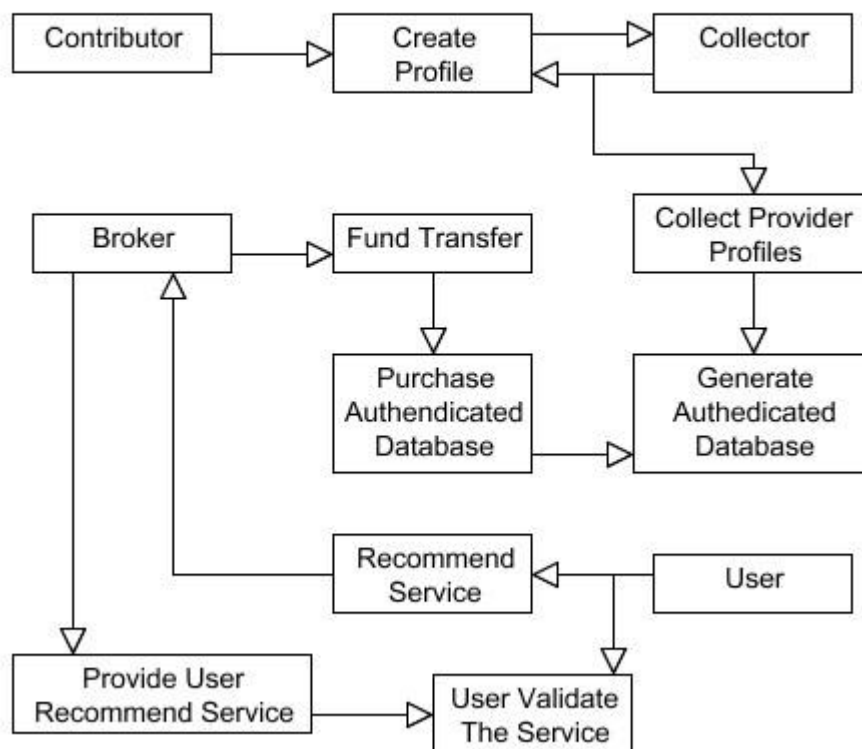
- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist.

The collaboration diagram illustrates messages being sent between classes and objects.



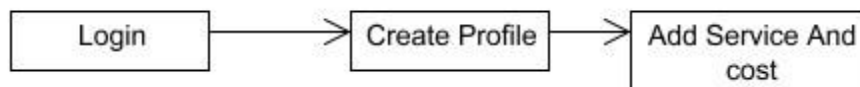
5.6 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

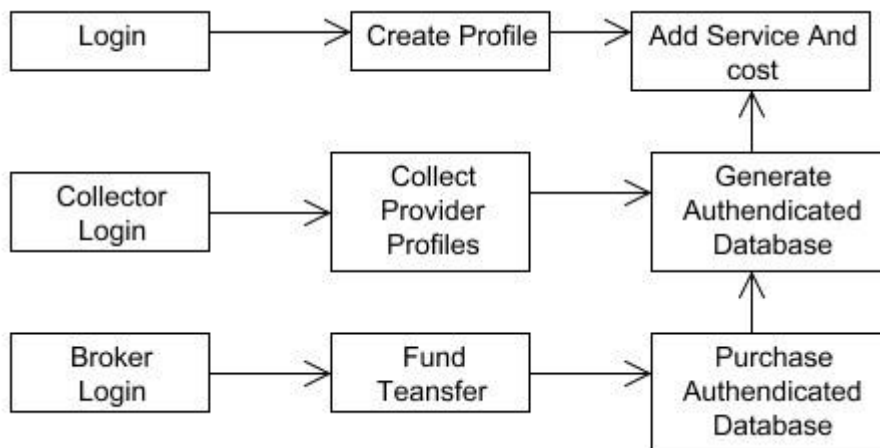
Level 0:



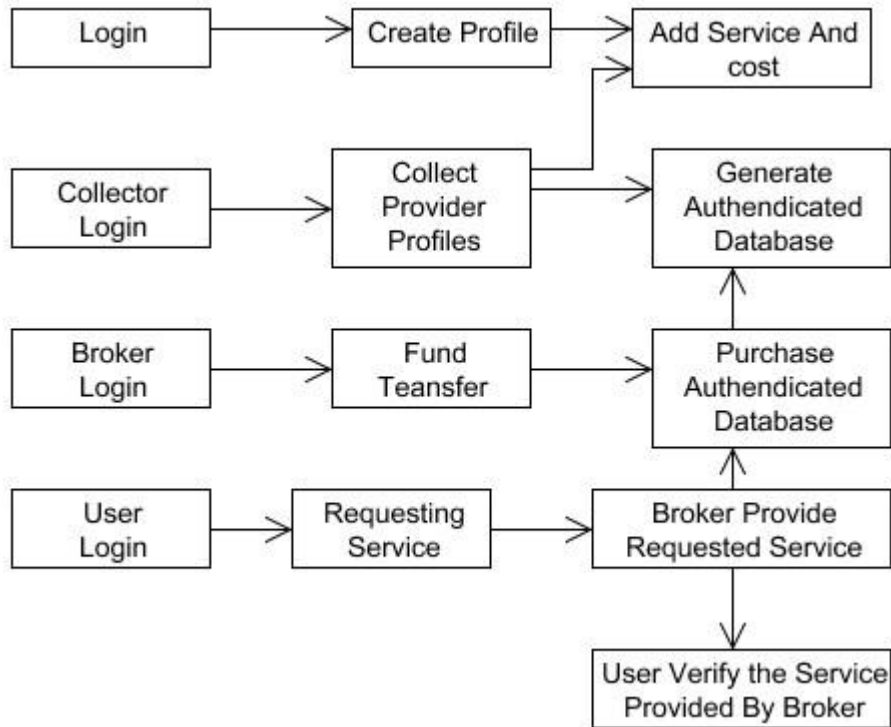
Level 1:



Level 2:



Level 3:



6.SYSTEM DESIGN

The architecture of the proposed model for data markets is designed as in Fig.1. The model consists of two layers: 1) Data Acquisition layer and 2) Data Trading layer. Apart from these layers four other entities also play a vital role in the system model namely data contributors, service providers, data consumers, and registration centre.

The data acquisition layer collects raw data from the data contributors through service providers. The registration centre is used to maintain an online database for various registrations that assigns a unique identity and password for each registered data contributor. The registration centre setups various other parameters such as the signature scheme and cryptosystem. Multiple registration centres are also set up which could be used when a single point of failure occurs in the model. The data trading layer is responsible for maintaining the truthfulness of the raw data collection.

It should be free of data collection attacks from external intruders and should properly verify whether the raw data provided is through registered users. The truthfulness of data processing in the data trading layer is used to represent the integrated correctness and completeness of the data processing results.

6.1 MODULES

- **Contributors Product Data & Pseudo identity Creation.**
- **Database Construction by the collector.**
- **Service Selection by the Service Provider.**
- **Results Verification by the Data Consumers.**

6.2 MODULE EXPLANATION:

6.2.1 Contributors Product Data & Pseudo Identity Creation

While the user undergoes registration process the Registration centre will provide the pseudo identity and provides them to the user. We assume that the registration centre sets up the system parameters at the beginning of data trading. The verification conducted by both the service provider and the data consumer. Between the two-layer batch verifications, we introduce data processing and signatures aggregation done by the service provider. At last, we present outcome verifications conducted by the data consumer. The data contributors is in need to expose the service that provided by them, in terms of the whole package of the service. The package that consists of the details such as a product that provide by the contributor and the respective cost for each product in service. And a total cost of the service. The data contributor can be able to produce any number (N numbers) of service and each are declared as separate package.

6.2.2 Database Construction by the collector

The collector surfs with the contributor services and select the needed package of services. And the collector submits the resource request to the respective CSP of service. If the

CSP acknowledge the collector request of resource, Now the collector is ready to access the resource details and to provide the respective resource to requesting service provider. Collector serves as an intermediate between the broker and the CSP.

6.2.3 Service Selection by the Service Provider

The service provider can be able to select the service that is in need from the service provided by the collector from the CSP. If the service provider selects their desired package of service, then the service provider should pay for the respective services. If the service provider is paid with the service the service provider can access the service from collector. And now the service provider is ready to outsource the service that bought from the collector to the user.

6.2.4 Results Verification by the Data Consumers The consumer search for the service that they need from the various service providers. And if the consumer finds the needed service, they request the service to the service provider and get use with the resource. And verify or cross check the resource that bought from the service provider that whether service provider serves the correct resource in affordable cost.

Results Verification by the Data Consumers

The consumer search for the service that they need from the various service providers. And if the consumer finds the needed service, they request the service to the service provider and get use with the resource. And verify or cross check the resource that bought from the service provider that whether service provider serves the correct resource in affordable cost.

7. CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalizing the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language.

Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

7.3 TEST PROCEDURE SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.4 TEST DATA AND OUTPUT

7.4.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design

(i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.4.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

7.4.3 PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

7.4.4 STRESS TEST

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

7.4.5 STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

7.4.6 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.5 TESTING TECHNIQUES / TESTING STRATEGIES

7.5.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one

that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

7.5.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- [Cyclometric complexity](#)
- Deriving test cases
- Graph matrices Control

7.5.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods

- Equivalence partitioning
- Boundary value analysis
- Comparison testing

7.5.2 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

7.5.2.1 INTEGRATION TESTING:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing.

There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

7.5.2.2 PROGRAM TESTING:

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

7.5.2.3 SECURITY TESTING:

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack.

During security, the tester places the role of individual who desires to penetrate system.

7.5.2.4 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- * The function or performance characteristics confirm to specifications and are accepted.
- * A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

7.5.2.5 USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.

This is done in regarding to the following points.

- Input screen design.
- Output screen design.

Source Code

ProvidingServices.java

```
package CloudServer;
import
java.io.BufferedReader;
r; import
java.io.FileReader;
import
com.opencsv.CSVRea
der; public class
ProvidingServices
{      public String profileprovide(String
recinfo)
    {
        String
recdata=recinfo;          String
response="";
        if(recdata.equalsIgnoreCase("RequestingProfile"))
        {
            String csvFile = "C:/CSPData/ProfileDetails.csv";
            BufferedReader br = null;
            String line = "";
            StringBuffer sb = new
StringBuffer();          try
            {
                CSVReader reader = new CSVReader(new
FileReader(csvFile));          br = new BufferedReader(new
FileReader(csvFile));          while ((line = br.readLine()) !=
null)
            {
```

```

        sb.append(line);
    sb.append("@");
    }
    response=removeLastIndex(sb.toString());
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
    }
    return response;
}
private static String removeLastIndex(String val)
{
    String values = val.substring(0, val.length() - 1);
    return values;
}
}

```

ProvidingServicesDelegate.java

```

package CloudServer;

@javax.jws.WebService(targetNamespace = "http://CloudServer/", serviceName =
"ProvidingServicesService", portName = "ProvidingServicesPort", wsdlLocation =
"WEB-INF/wsdl/ProvidingServicesService.wsdl")
public class ProvidingServicesDelegate {

    CloudServer.ProvidingServices providingServices = new
CloudServer.ProvidingServices();

    public String profileprovide(String recinfo) {
        return
providingServices.profileprovide(recinfo);
    }

}

```

DBconnection.java

```
package DB;

import java.sql.Connection;
import java.sql.DriverManager;

import javax.servlet.ServletContext;
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;

public class DBconnection implements
ServletContextListener {
    ServletContext;
    Connection connection=null;
    // Connection connectionpro=null;
    public void contextDestroyed(ServletContextEvent sce)
    {

        try
        {
            connection.close();
            // connectionpro.close();
            System.out.println("=====Connection Closed Successfully
For Data Contributors=====");
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }

    }

    public void contextInitialized(ServletContextEvent sce)
    {
        servletcontext=sce.getServletContext();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
```

```

connection=DriverManager.getConnection("jdbc:mysql://localhost/datacontributors",
"root","root");
//
        connectionpro=DriverManager.getConnection("jdbc:mysql://localhost/", "root", "root")
;

servletcontext.setAttribute("connection",connection); //
servletcontext.setAttribute("connectionpro",connectionpro);
        System.out.println("=====Connection Created
Successfully For Data Contributors=====");
    }
    catch (Exception e)
    {
        e.printStackTrace ();
    }
}
public Connection getConnection ()
{
    return connection;
}
// public Connection getConnection1()
// {
//     return connectionpro;
// }
}

```

DBStatement.java

```

package DB;
import java. Sql. Connection;

import javax.servlet.ServletContext;

public class DBStatement
{
    Connection connection=null;
    ServletContext context;
    String sql="";
    DBconnection db=new DB connection();

    public Connection getConnection ()
    {

```

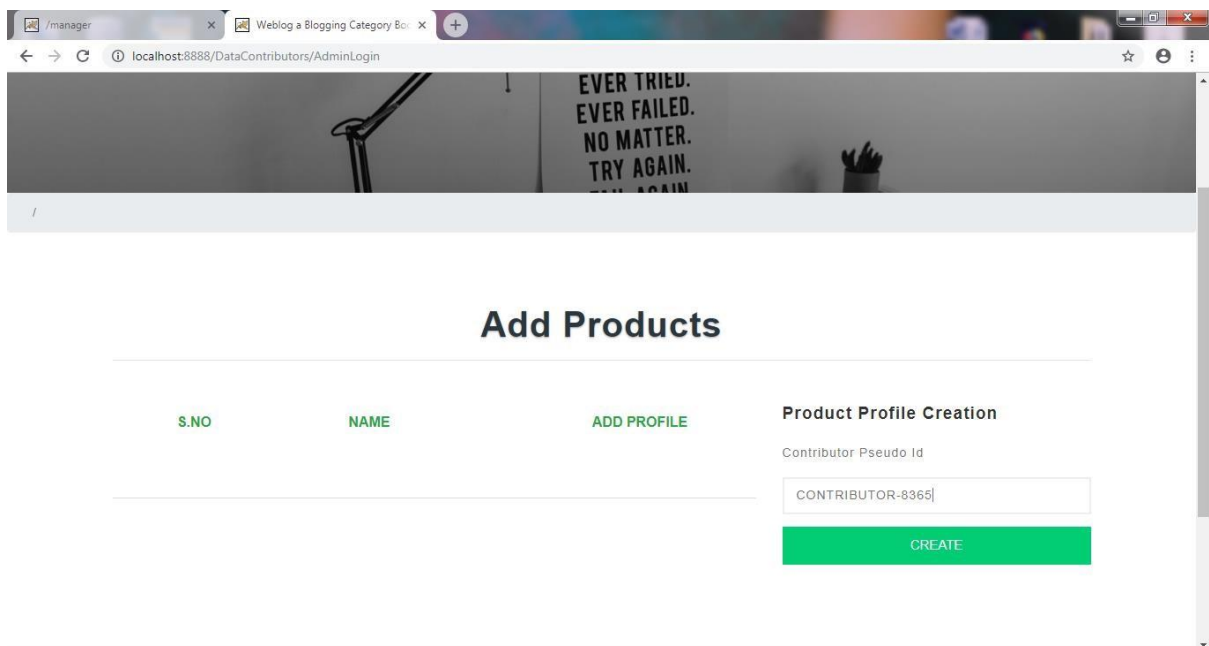
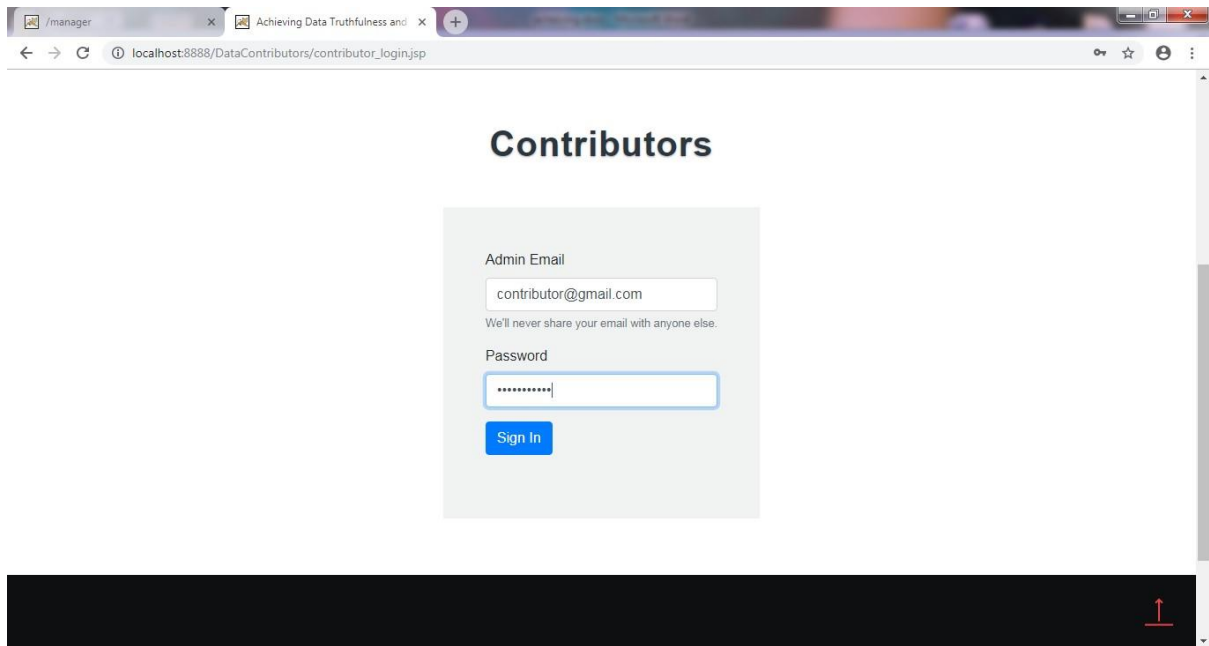
```

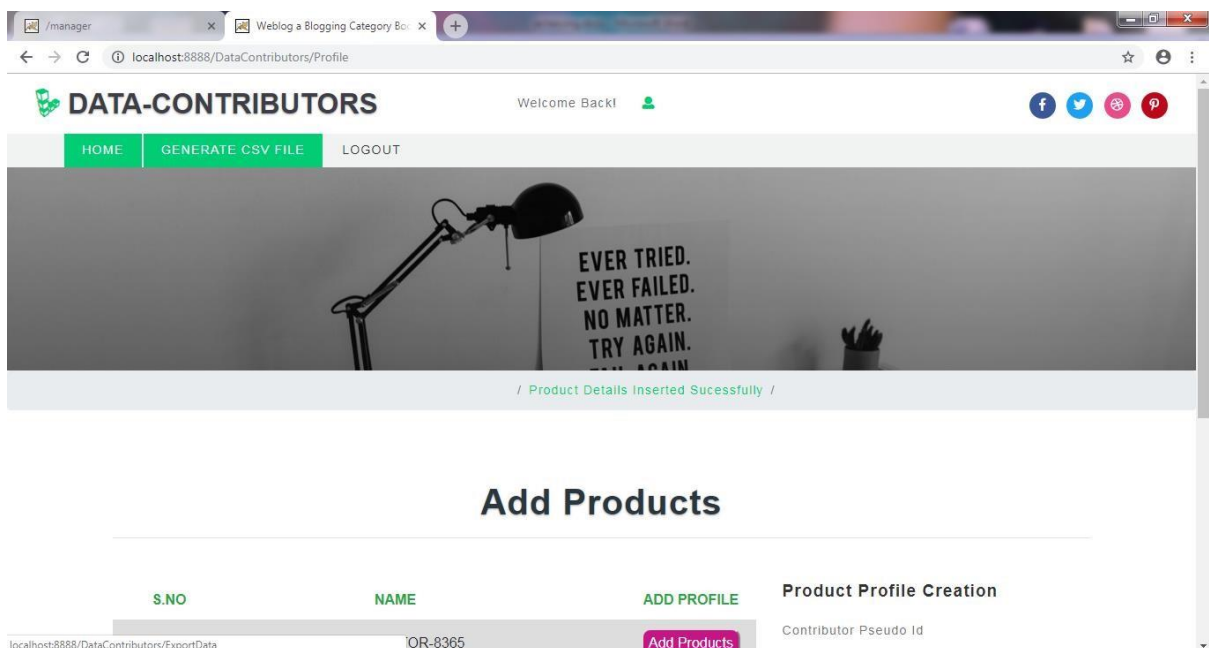
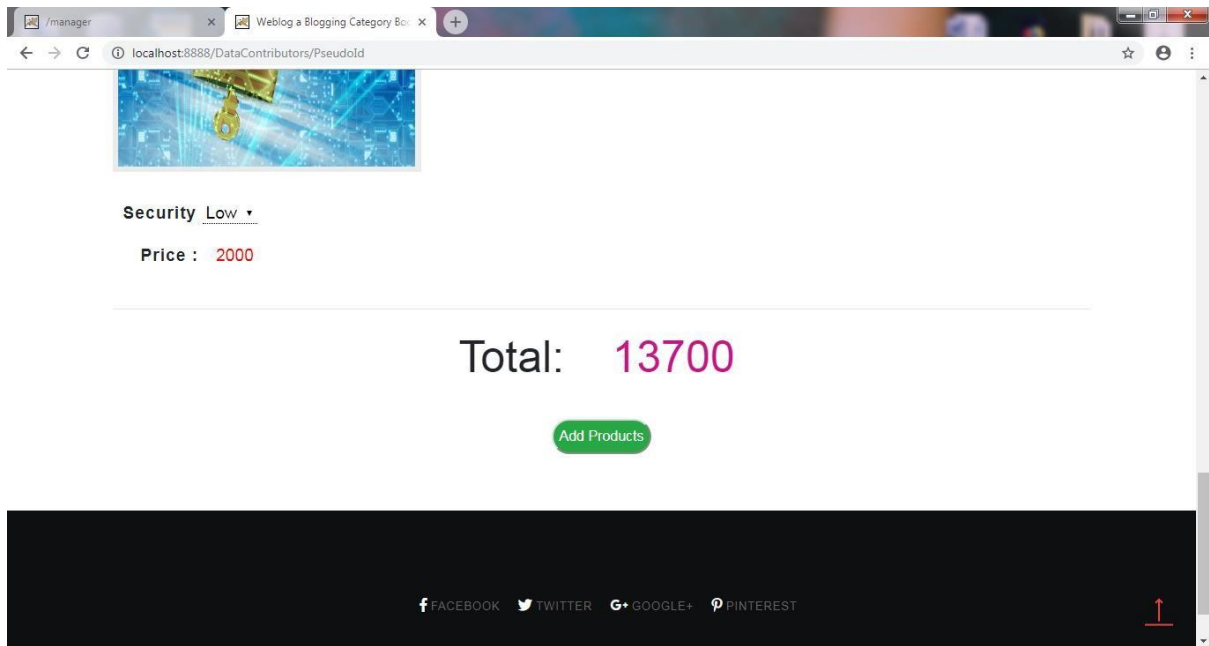
connection=db.getConnection();
return connection;
    }
    public String insertpname()
    {
        sql="insert into contributors values(?,?,?,?)";
        return sql;
    }
    public String insert provider()
    {
        sql="insert into contributors data values(?,?,?,?,?,?,?,?,?)";
        return sql;
    }

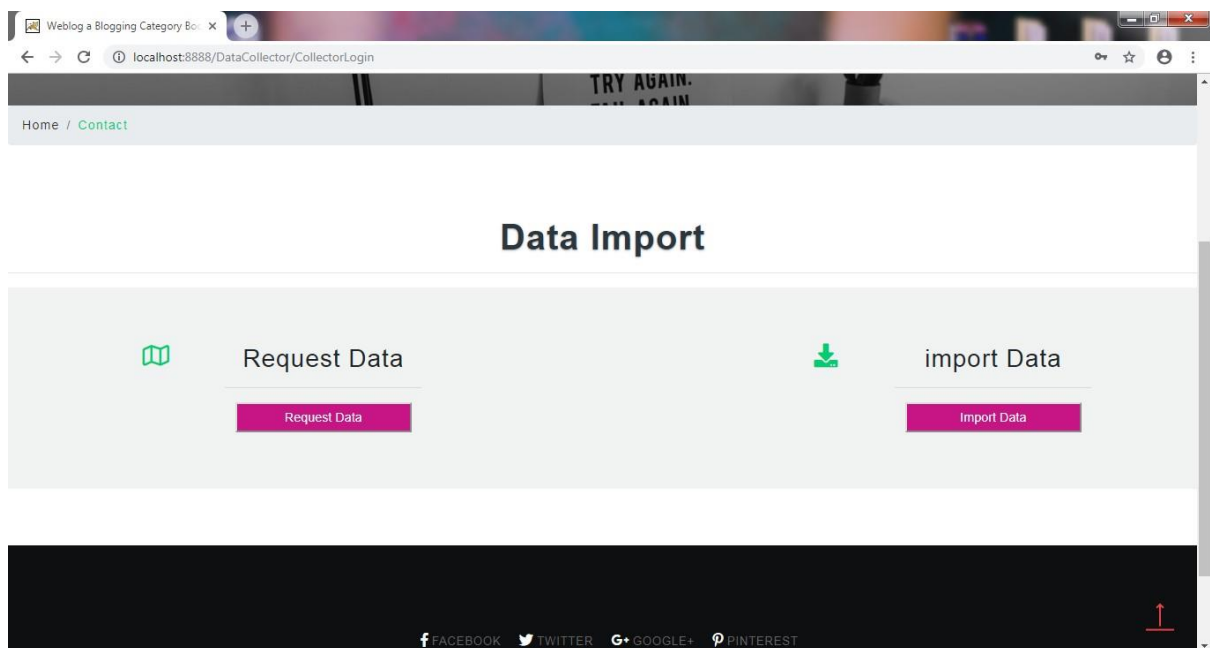
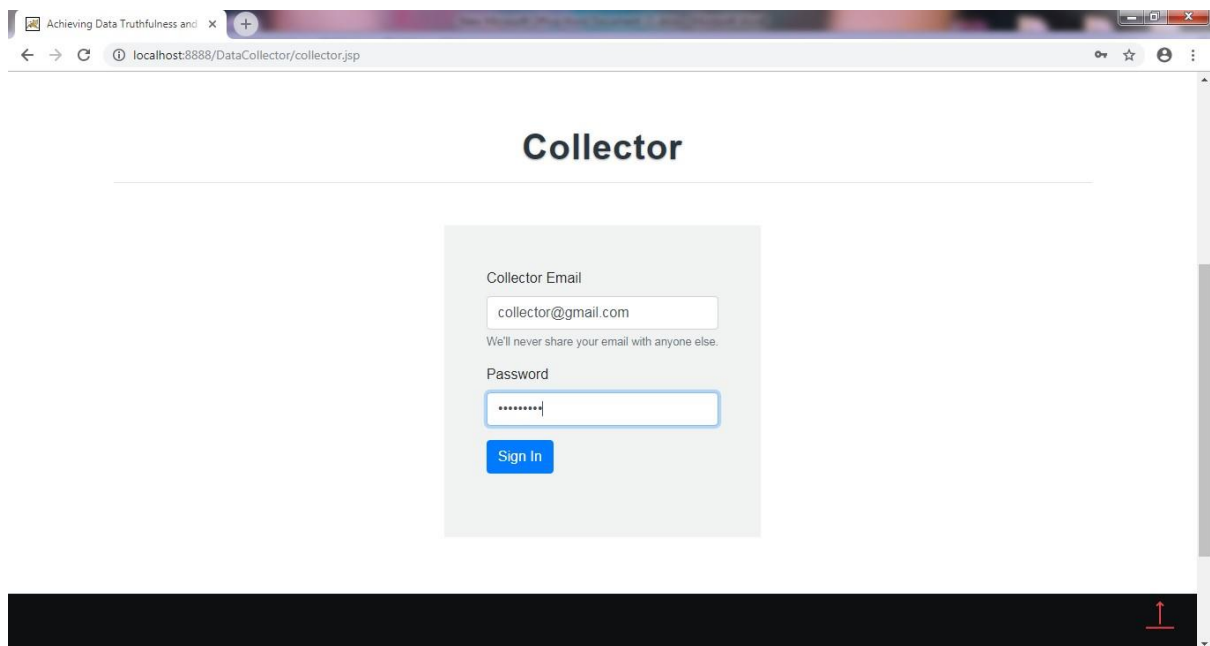
    public String updatestatus ()
    {
        sql="update contributors set status=? where
contributor_id=?";
        return sql;
    }
    public String checkprofile ()
    {
        sql="select * from contributors where
contributor_id=?";
        return sql;
    }
    public String updateProvider ()
    {
        sql="update contributors set permission=?export=? where
contributor_id=?";
        return sql;
    }
}

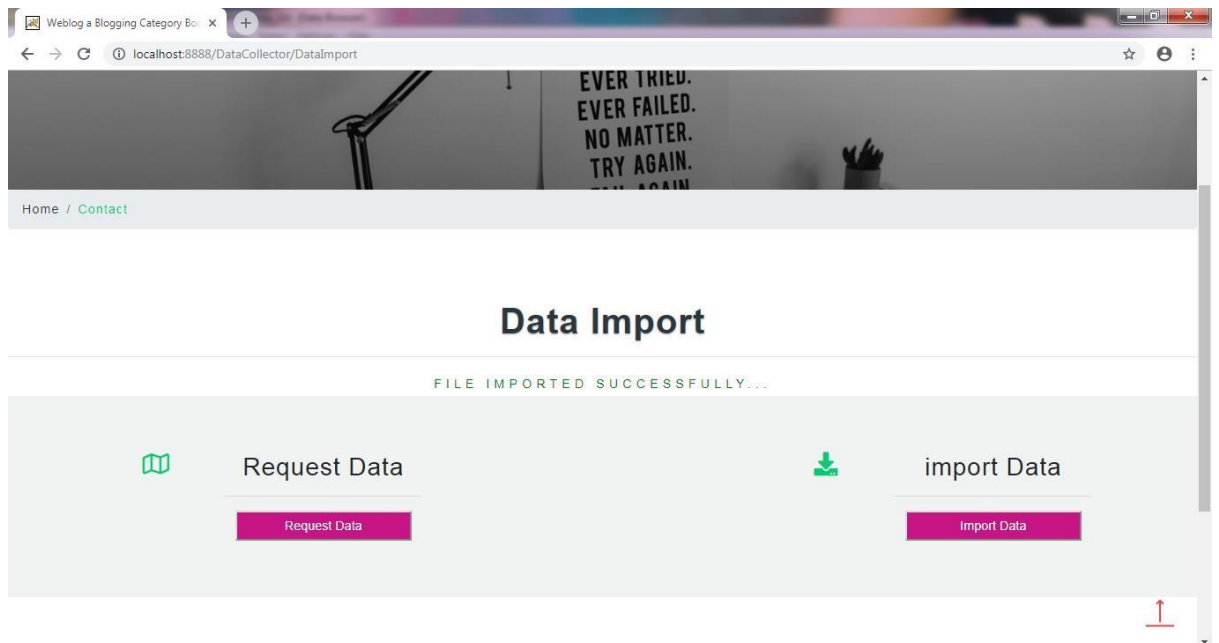
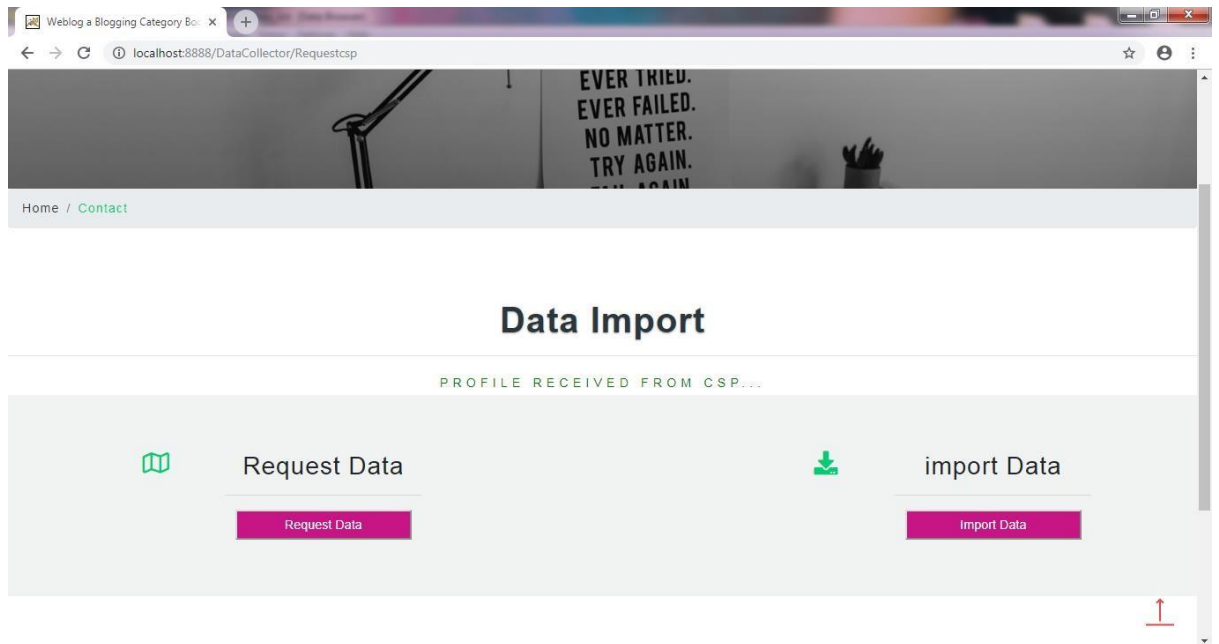
```

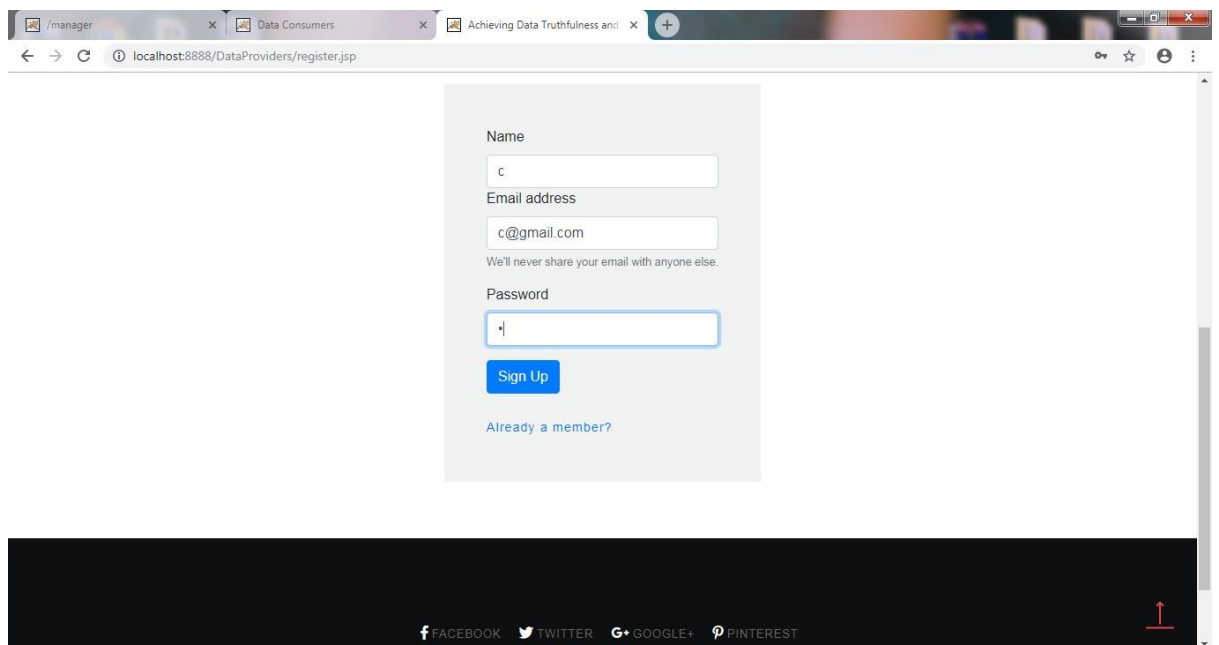
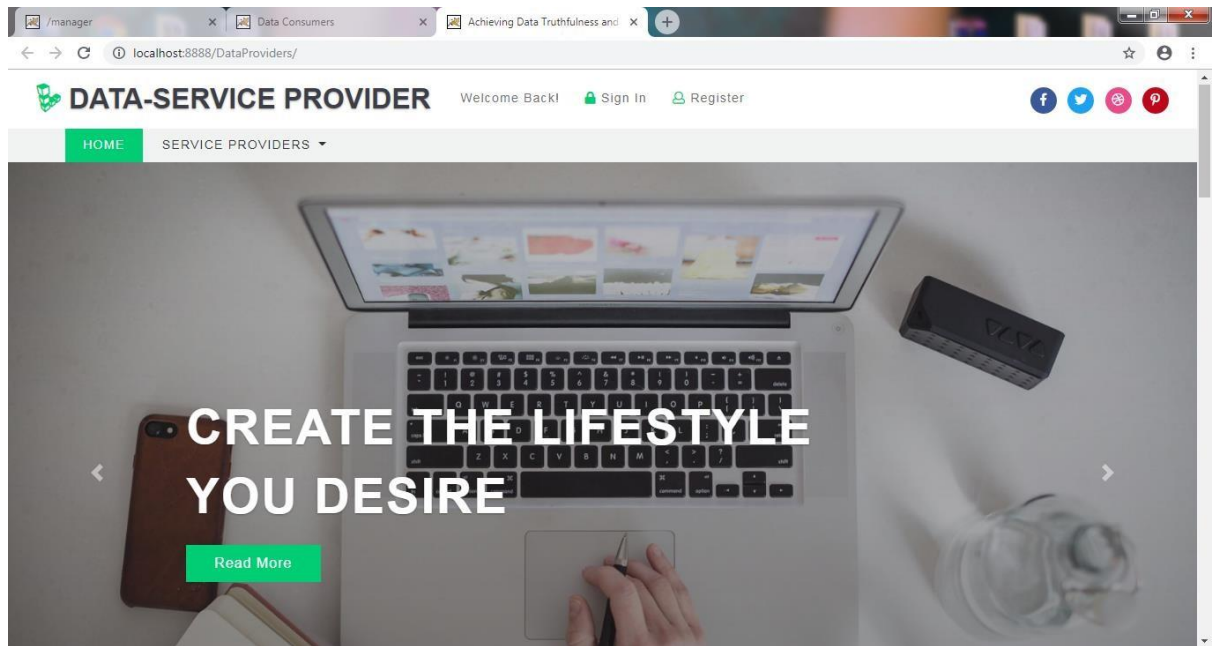
SNAP SHOTS

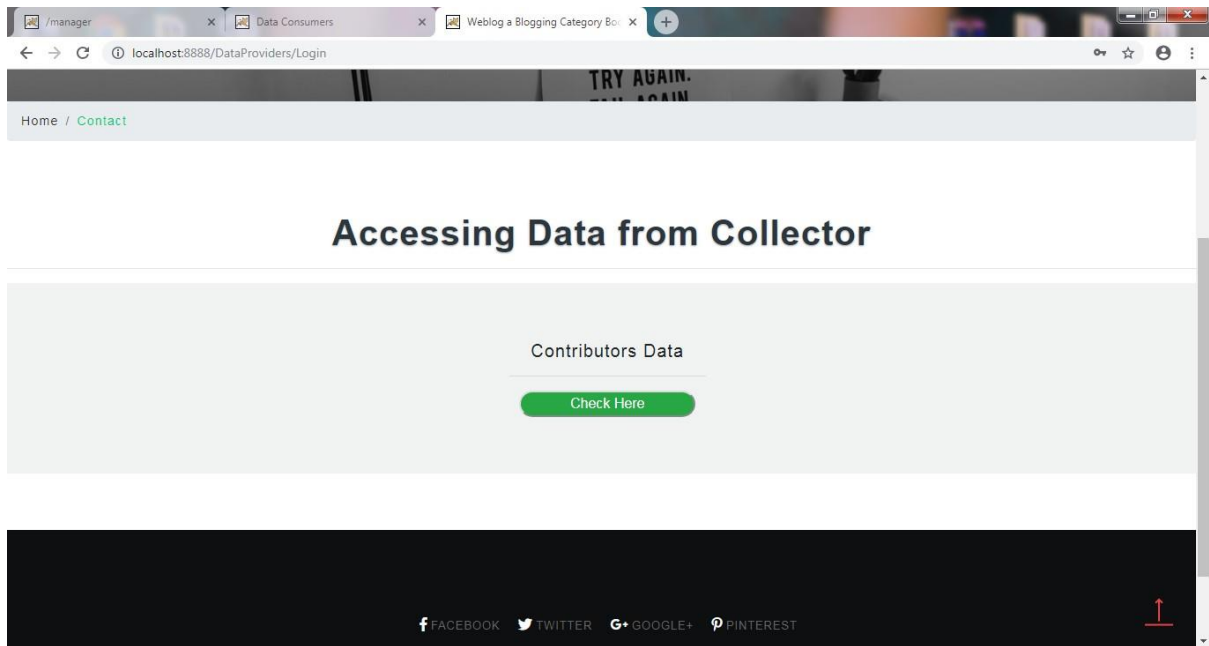












S.No	Id	Provider Name	No.of Systems	Operating System	Processor	RAM	Server	Data Base	Mother Board	Security	Amount
1.	s1	CONTRIBUTOR-2045	1	Windows7	IntelCoreI3	2GBRAM	Tomcat6	MySQL	Asus	High	15000
2.	s2	CONTRIBUTOR-2152	2	Windows8	IntelCoreI5	4GBRAM	Tomcat7	Oracle	Gigabyte	Low	13300
3.	s3	CONTRIBUTOR-2045	3	Windows10	IntelCoreI7	8GBRAM	Tomcat7	MySQL	ASRock	High	15500

Total amount for all products is **43800** /-

You have a discount of **10 %** profit for purchasing a Database.

Finally you have to Pay Rs. **39420** /- to Purchase this Database From Collector.

Rs. **43800** /- 10 % discount.

Rs **39420** /-

[Pay Now](#)

government tender sample - G... | manohar - Lenovo A7010a48 - V... | Bank Server

localhost:8888/BankAdmin/banklogin

**PUBLIC BANK**

HOME NEWUSER ADDAMOUNT LOGOUT

New Account Holder

Account Holder Name:

Mobile Number:

E-mail:

Address:

City:

PinCode:

State:

©Google zone. All right reserved.

/manager | Data Consumers | Bank Server

Not secure | 10.0.0.14:8888/BankAdmin/setuserpayment.jsp?sessionId=SD682389081A82E387A90DF0E6AF9EA8

10.0.0.14:8888 says

Amount Debited From your account...

HOME CONTACT

Payment

Account No:

Atm Pin:

OPass:

©Google zone. All right reserved.

Browser tabs: /manager, Achieving Data Truthfulness and...

Address bar: localhost:8888/DataConsumers/register.jsp

Name

Email address

We'll never share your email with anyone else.

Password

Mobile

Address

Sign Up

[Already a member?](#)

Browser tabs: /manager, Data Consumers

Address bar: localhost:8888/DataConsumers/Login

List of products based on your requirements

Broker Names ☐ a ☒ b

Operating System

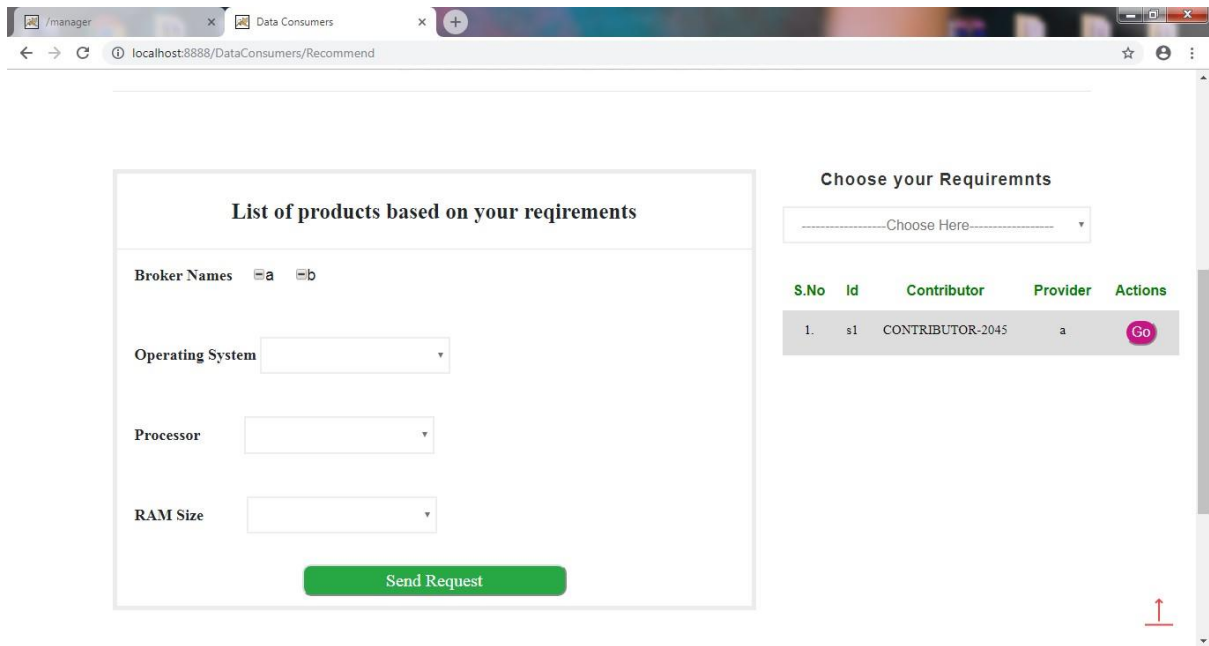
Processor

RAM Size

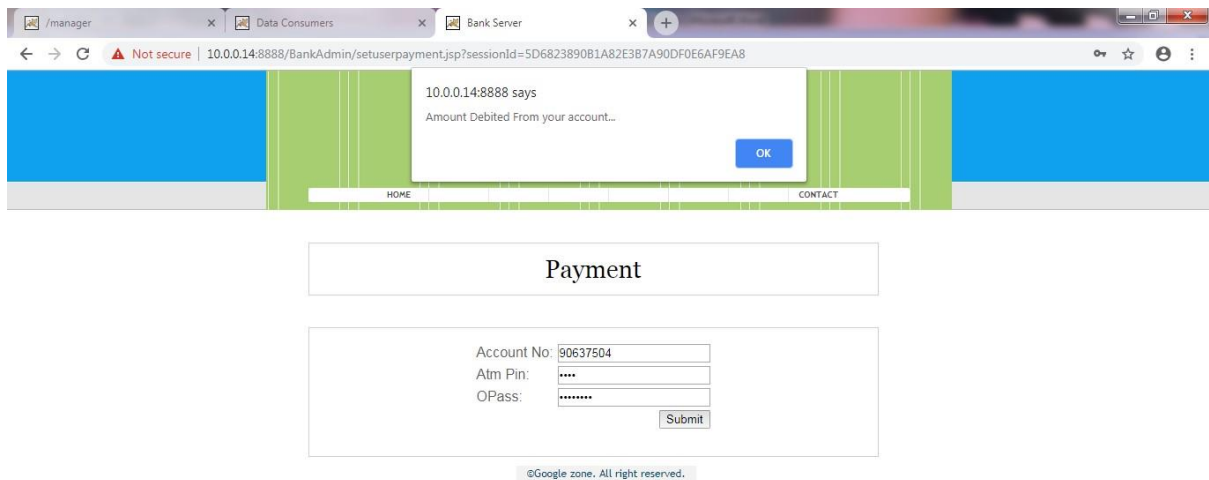
Send Request

Choose your Requirements

Choose Here



Bank Screenshot



REFERENCES

1. G. Eason, List of Social Networking Websites 2009
http://en.wikipedia.org/wiki/List_of_social_networking_websites
2. Xu, L., Jiang, C., Wang, J., Yuan, J., & Ren, Y. (2014). Information security in big data: privacy and data mining. *IEEE Access*, 2, 1149-1176.
3. Ren, K., Lou, W., Kim, K., & Deng, R. (2006). A novel privacy preserving authentication and access control scheme for pervasive computing environments. *IEEE Transactions on Vehicular Technology*, 55(4), 1373-1384.
4. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4), 469-472.
5. Ghinita, G., Kalnis, P., & Tao, Y. (2011). Anonymous publication of sensitive transactional data. *IEEE Transactions on Knowledge and Data Engineering*, 23(2), 161-174.
6. Balazinska, M., Howe, B., & Suciu, D. (2011). Data markets in the cloud: An opportunity for the database community. *Proc. of the VLDB Endowment*, 4(12), 1482-1485.
7. Patil Maulik, Y., & Yeola, M. (2015). Generating Private Recommendations Using ElGamal Homomorphic Encryption. *International Journal Of Engineering And Computer Science*, 4(06).
8. Polat, H., & Du, W. (2003, November). Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE*

- International Conference on* (pp. 625-628). IEEE.
9. Polat, H., & Du, W. (2005). Privacy-preserving collaborative filtering. *International journal of electronic commerce*, 9(4), 9-35. McSherry, F., & Mironov, I. (2009, June). Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD*
 10. *international conference on Knowledge discovery and data mining* (pp. 627-636). ACM.
 11. Cissée, R., & Albayrak, S. (2007, May). An agent-based approach for privacy-preserving recommender systems. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems* (p. 182). ACM.
 12. Canny, J. (2002). Collaborative filtering with privacy. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on* (pp. 45-57). IEEE.
 13. Erkin, Z., Veugen, T., Toft, T., & Lagendijk, R. L. (2012). Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE transactions on information forensics and security*, 7(3), 1053-1066.
 14. Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), 734-749.
 15. Popa, R. A., Blumberg, A. J., Balakrishnan, H., & Li, F. H. (2011, October). Privacy and accountability for location-based aggregate statistics. In *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 653-666). ACM.

16. Chapelle, O., & Chang, Y. (2011, January). Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge* (pp. 1-24).
17. 2009 RECS Dataset,”
<https://www.eia.gov/consumption/residential/data/2009/index.php?view=microdata>.
18. “PBC Library,” <https://crypto.stanford.edu>ⁱ