



JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

User Privacy Conservation Over Cloud To Eradicate Shortcoming Occurrence

S. Pravin, Dr.V.Kavitha
MCA Student, Professor
Hindusthan College of Arts and Science (Autonomous)

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Line shopping is becoming more and more common in our daily lives. Understanding users' interests and behavior is essential in order to adapt e-commerce websites to customers' requirements. The information about users' behavior is stored in the web server logs. The analysis of such information has focused on applying data mining techniques where a rather static characterization is used to model users' behaviour and the sequence of the actions performed by them is not usually considered. Therefore, incorporating a view of the process followed by users during a session can be of great interest to identify more complex behavioral patterns. To address this issue, this paper proposes a linear-temporal logic model checking approach for the analysis of structured e-commerce web logs. By defining a common way of mapping log records according to the e-commerce structure, web logs can be easily converted into event logs where the behaviour of users is captured. Then, different predefined queries can be performed to identify different behavioural patterns that consider the different actions performed by a user during a session. Finally, the usefulness of the proposed approach has been studied by applying it to a real case study of a Spanish e-commerce website. The results have identified interesting findings that have made possible to propose some improvements in the website design with the aim of increasing its efficiency.

1.2 MODULE DESCRIPTION

Admin

User

Intermediate

Admin

Login

In this module admin having a unique user name and password for a login session admin having a all the rights to do any change son our web application they will monitoring all the process on our web application.

View

In this session admin can view all of the users and analyst details on this module, if the admin view user's personal information who enters on new to the user data privacy e commerce site web application.

User

Register & login

If the user first time enters in our web application they will enters all of the personal information for a registration process after they will complete a registration they having a unique user name and password for the login session.

Search product

In this model user will search products on a e commerce site project. It's an uses like a usual e commerce site web application. All of the products available on this web application. Intermediates

The entire e commerce web sites having this types of users. Data analyst will analyze a user data details on this web application sometimes users information will sell on another e commerce web site so that is a major works on inter mediator.

2 SYSTEM STUDY

2.1 EXISTING SYSTEM

An e-commerce website is an open system where almost any customer behavior is possible. Current generation depends on the online shopping system. Online shopping users will search and buy products through via online. But in-depth users data will be stored on the server.

DISADVANTAGES

Users' data will lose on easily.

They did not provide a data secured on users.

2.2 PROPOSED SYSTEM

In this paper we propose the use of Temporal Logic and model checking techniques as an alternative to data mining techniques. Such techniques have proved their applicability for open systems. We propose here a methodology for using it in structured e-commerce websites. The goal is to analyze the usage of ecommerce websites and to discover customers' complex behavioral patterns by means of checking temporal logic formulas describing such behaviors against the log model.

ADVANTAGES

- Provide a security for every user's data.
- Using some new methodology for protect user data.
- We implement hashing algorithm for more secured process.

3 SOFTWARE SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- Processor : Dual core processor 2.6.0 GHz
- RAM : 1GB
- Hard disk : 160 GB
- Compact Disk : 650 MB
- Keyboard : Standard keyboard
- Monitor : 15 inch color monitor

3.2 SOFTWARE REQUIREMENTS

- Front End : PHP
- Back End : My SQL
- Platform : Windows 7

PHP

PHP is server side back end programming language. It executes in server along with maximum all available web servers like Apache, IIS (Internet Information Server) etc..., and return the response as required MIME type. It is a Pre Process Hypertext, we could do many things on server by using PHP on server and co-ordinate with DB server for CURD (Create, Update, Read, and Delete) actions. Front end in the seance, UI which intact the users, it can done by HTML, or any others. And UI Behavior is defined in UI back end Languages (Scripting languages) via: Java script, VB script

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common Uses of PHP

PHP performs system functions, i.e. from files on a system it can create, open, read, write and close them. The other uses of PHP are:

PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user. You add, delete and modify elements within your database thru PHP. Access cookies variables and set cookies. Using PHP, you can restrict users to access some pages of your website. It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP Variables

The main way to store information in the middle of a PHP program is by using a variable. Here are the most important things to know about variables in PHP.

- All variables in PHP are denoted with a leading dollar sign (\$).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.

PHP variables are Perl-like. PHP has a total of eight data types which we use to construct our variables:

- **Integers:** are whole numbers, without a decimal point, like 4195.
- **Doubles:** are floating-point numbers, like 3.14159 or 49.1.
- **Booleans:** have only two possible values either true or false.
- **NULL:** is a special type that only has one value: NULL.
- **Strings:** are sequences of characters, like 'PHP supports string operations.'
- **Arrays:** are named and indexed collections of other values.
- **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
- **Resources:** are special variables that hold references to resources external to PHP (such as database connections).

Back End (MySQL)

MySQL is the world's most used open source relational database management system (RDBMS) as of 2008 that run as a server providing multi-user access to a number of databases. The MySQL development project

has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack—LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, Word Press, phpBB, MyBB, Drupal and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), ImagebookTwitter, Flickr, Nokia.com, and YouTube.

Inter images

MySQL is primarily an RDBMS and ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

Graphical

The official MySQL Workbench is a free integrated environment developed by MySQL AB, which enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL frontend, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

MySQL ships with some command line tools. Third-parties have also developed tools to manage a MySQL server, some listed below. Maatkit - a cross-platform toolkit for MySQL, PostgreSQL and Memcached, developed in Perl Maatkit can be used to prove replication is working correctly, fix corrupted data, automate

repetitive tasks, and speed up servers. Maatkit is included with several GNU/Linux distributions such as CentOS and Debian and packages are available for Programming. MySQL works on many different system platforms, including AIX, BSDi, FreeBSD, HP-UX, eComStation, i5/OS, IRIX, Linux, Mac OS X, Microsoft Windows, NetBSD, Novell NetWare, OpenBSD, OpenSolaris, OS/2 Warp, QNX, Solaris, Symbian, SunOS, SCO Open Server, SCO UnixWare, Sanos and Tru64. A port of MySQL to OpenVMS also exists. MySQL is written in C and C++. Its SQL parser is written in yacc, and a home-brewed lexical analyzer. Many programming languages with language-specific APIs include libraries for accessing MySQL databases. These include MySQL Connector/Net for integration with Microsoft's Visual Studio (languages such as C# and VB are most commonly used) and the JDBC driver for Java. In addition, an ODBC interimage called MyODBC allows additional programming languages that support the ODBC inter image to communicate with a MySQL database, such as ASP or ColdFusion. The HTSQL - URL-based query method also ships with a MySQL adapter, allowing direct interaction between a MySQL database and any web client via structured URLs.

Features

As of April 2009, MySQL offered MySQL 5.1 in two different variants: the open source MySQL Community Server and the commercial Enterprise Server. MySQL 5.5 is offered under the same licenses. They have a common code base and include the following features:

A broad subset of ANSI SQL 99, as well as extensions

- Cross-platform support
- Stored procedures
- Triggers
- Cursors
- Updatable Views
- Information schema

Strict mode (ensures MySQL does not truncate or otherwise modify data to conform to an underlying data type, when an incompatible value is inserted into that type)

X/Open XAdistributed transaction processing (DTP) support; two phase commit as part of this, using Oracle's InnoDB engine

- Transactions with the InnoDB, and Cluster storage engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)

- Replication support (i.e. Master-Master Replication & Master-Slave Replication)
- Embedded database library
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Hot backup (via `mysqlhotcopy`) under certain conditions

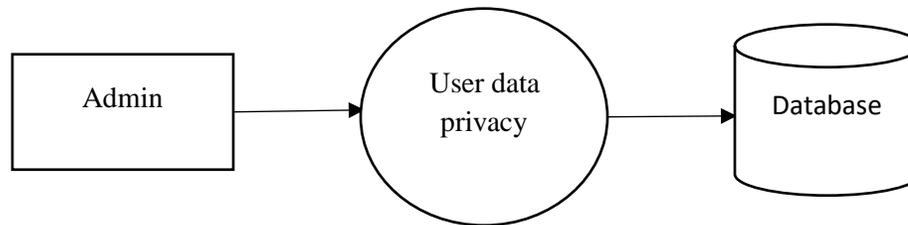
Multiple storage engines, allowing one to choose the one that is most effective for each table in the application (in MySQL 5.0, storage engines must be compiled in; in MySQL 5.1, storage engines can be dynamically loaded at run time): Native storage engines (MyISAM, Falcon, Merge, Memory (heap), Federated, Archive, CSV, Black hole, Cluster, EXAMPLE, Maria, and InnoDB, which was made the default as of 5.5). Partner-developed storage engines (solidDB, NitroEDB, ScaleDB, TokuDB, Infobright (formerly Brighthouse), Kickfire, XtraDB, IBM DB2). InnoDB used to be a partner-developed storage engine, but with recent acquisitions, Oracle now owns both MySQL core and InnoDB.

4. SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers. A context diagram gives an overview and it is the highest level in a data flow diagram, containing only one process representing the entire system. It should be split into major processes which give greater detail and each major process may further split to give more detail. Level 0 DFD must balance with the context diagram it describes. Input going into a process is different from outputs leaving the process. Data stores are first shown at this level.

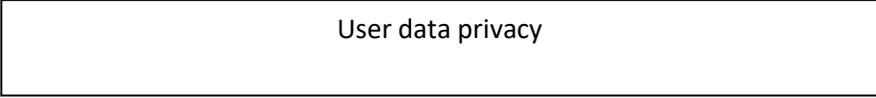


LEVEL 1

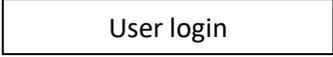
DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. Level 1 - interaction between 2 different business applications. This is primarily used to explain the process to business and tech leads, QA leads. As described previously, context diagrams (level 0 DFDs) are diagrams where the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

The next stage is to create the Level 1 Data Flow Diagram. This highlights the main functions carried out by the system. As a rule, to describe the system was using between two and seven functions - two being a simple system and seven being a complicated system. This enables us to keep the model manageable on screen or paper

4.2 INPUT DESIGN



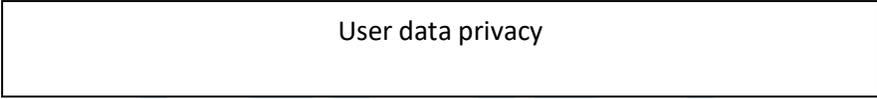
User data privacy



User login

| | |
|-----------|----------|
| User name | |
| password | |
| | New user |

4.3 OUTPUT DESIGN



User data privacy



User login

| | |
|-----------|----------|
| User name | Pravin |
| password | ***** |
| | New user |

4.4 TABLE DESIGN

add_product

PRIMARY KEY: Id

| Field | Type | Null | Default |
|----------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| product | varchar(100) | Yes | NULL |
| ptype | varchar(100) | Yes | NULL |
| brand | varchar(100) | Yes | NULL |
| model | varchar(100) | Yes | NULL |
| price | varchar(100) | Yes | NULL |
| quantity | int(11) | Yes | NULL |
| image | varchar(100) | Yes | NULL |
| uname | varchar(100) | Yes | NULL |
| count | varchar(100) | Yes | NULL |
| rdate | varchar(100) | Yes | NULL |

Admin

PRIMARY KEY: Id

| Field | Type | Null | Default |
|----------|--------------|------|---------|
| utype | varchar(100) | Yes | NULL |
| username | varchar(100) | Yes | NULL |
| password | varchar(100) | Yes | NULL |

analyst_reg

PRIMARY KEY: Id

| Field | Type | Null | Default |
|---------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| name | varchar(100) | Yes | NULL |
| contact | bigint(20) | Yes | NULL |
| email | varchar(100) | Yes | NULL |
| otp | int(11) | Yes | NULL |
| rdate | varchar(100) | Yes | NULL |

Payment

PRIMARY KEY: Id

| Field | Type | Null | Default |
|-------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| uname | varchar(100) | Yes | NULL |
| accno | int(11) | Yes | NULL |
| card | int(11) | Yes | NULL |
| vdate | varchar(100) | Yes | NULL |
| rdate | varchar(100) | Yes | NULL |

purchase_details

PRIMARY KEY: Id

| Field | Type | Null | Default |
|---------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| uname | varchar(100) | Yes | NULL |
| product | varchar(100) | Yes | NULL |

| | | | |
|------------|--------------|-----|------|
| ptype | varchar(100) | Yes | NULL |
| model | varchar(100) | Yes | NULL |
| price | int(11) | Yes | NULL |
| quantity | int(11) | Yes | NULL |
| tot_amount | int(11) | Yes | NULL |
| image | varchar(100) | Yes | NULL |
| rdate | varchar(100) | Yes | NULL |

Register

PRIMARY KEY: Id

| Field | Type | Null | Default |
|---------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| name | varchar(100) | Yes | NULL |
| contact | bigint(20) | Yes | NULL |
| email | varchar(100) | Yes | NULL |
| gender | varchar(100) | Yes | NULL |
| age | int(11) | Yes | NULL |
| address | varchar(100) | Yes | NULL |
| uname | varchar(100) | Yes | NULL |
| pass | varchar(100) | Yes | NULL |
| rdate | varchar(100) | Yes | NULL |

search_count

PRIMARY KEY: Id

| Field | Type | Null | Default |
|-------------|--------------|------|---------|
| id | int(11) | Yes | NULL |
| uname | varchar(100) | Yes | NULL |
| product | varchar(100) | Yes | NULL |
| public_key | varchar(100) | Yes | NULL |
| private_key | varchar(100) | Yes | NULL |
| encrypted | varchar(100) | Yes | NULL |
| status | varchar(100) | Yes | NULL |

4.5 INPUT DESIGN

Input design is the process of converting the user-oriented. Input to a computer based format. The goal of the input design is to make the data entry easier , logical and free error. Errors in the input data are controlled by the input design. The quality of the input determines the quality of the system output. All the data entry screen are interactive in nature, so that the user can directly enter into data according to the prompted messages. The user are also can directly enter into data according to the prompted messages. The users are also provided with option of selecting an appropriate input from a list of values. This will reduce the number of error, which are otherwise likely to arise if they were to be entered by the user itself. Input design is one of the most important phase of the system design. Input design is the process where the input received in the system are planned and designed, so as to get necessary information from the user, eliminating the information that is not required. The aim of the input design is to ensure the maximum possible levels of accuracy and also ensures that the input is accessible that understood by the user. The input design is the part of overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify the errors.

The objectives considered during input design are

- Nature of input processing.
- Flexibility and thoroughness of validation rules.
- Handling of properties within the input documents.
- Screen design to ensure accuracy and efficiency of the input relationship with files.
- Careful design of the input also involves attention to error handling, controls, batching and validation procedures.

Input design features can ensure the reliability of the system and produce result from accurate data or they can result in the production of erroneous information.

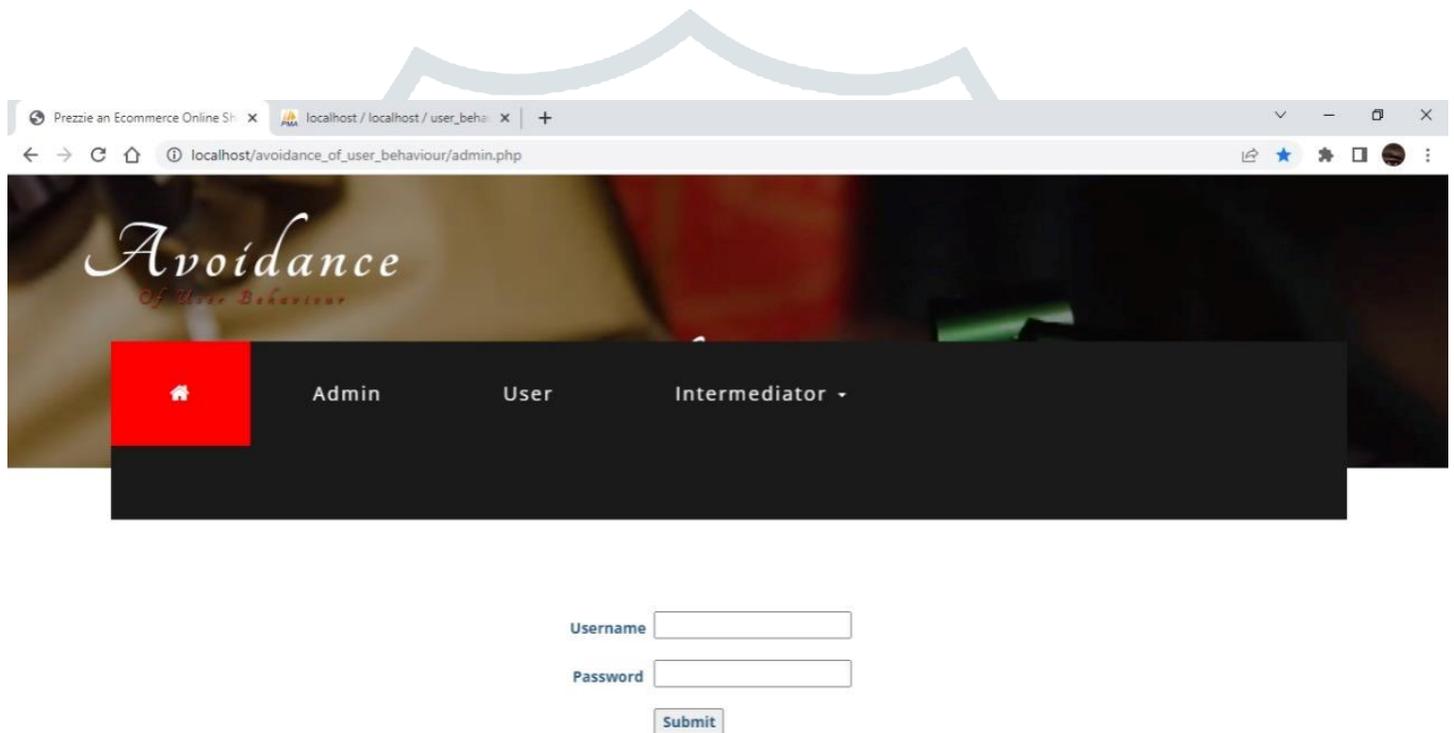


Figure: 4.5.1 Login Page Website

The Login page website has the login details for the Admin add products see a user details and purchase details. The Admin was give products details and can see user which prouduct overcome in items.

4.6 OUTPUT DESIGN

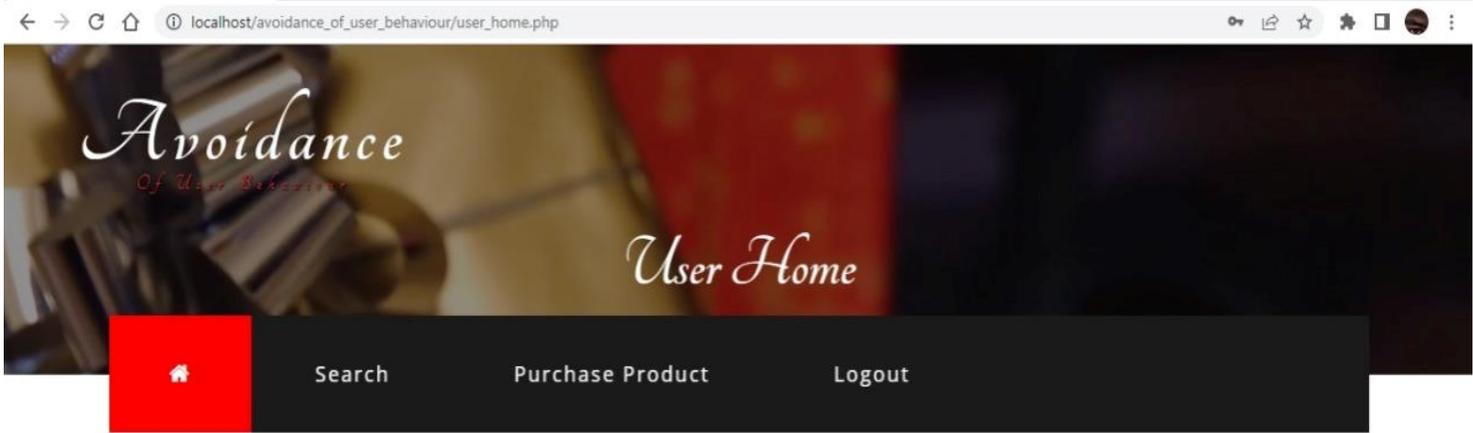
The output form of the system is either by screen or by hard copies. Output design aims at communicating the results of the processing of the users. The reports are generated to suit the needs of the users. The reports have to be generated with appropriate levels. In our project outputs are generated by asp as html pages. As its web application output is designed in a very user-friendly this will be through screen most of the time.

CODE DESIGN

The main purpose of code design is to simplify the coding and to achieve better performance and quality with free of errors. The coding is prepared in such a way that the internal procedures are more meaningful validation manager is displayed for each column. The coding of the variables is done in such a way that one other than person who developed the packages can understand its purpose. To reduce the server load, the project is designed in a way that most of the Validation of fields is done as client side validation, which will be more effective.

DATABASE DESIGN

The database design involves creation of tables that are represented in physical database as stored files. They have their own existence. Each table constitute of rows and columns where each row can be viewed as record that consists of related information and column can be viewed as field of data of same type. The table is also designed with some position can have a null value. The database design of project is designed in such a way values are kept without redundancy and with normalized format.



Welcome : suriya

© 2020 Prezzie. All rights reserved | Design by Admin

Figure: 4.5.2 User Details Page

The User Details page shows the users which products will purchase can see that details. And search some products then purchase.

5.SYSTEM TESTING AND IMPLEMENTATION

5.1 System testing

After a system has been verified, it needs to be thoroughly tested to ensure that every component of the system is performing in accordance with the specific requirements and that it is operating as it should including when the wrong functions are requested or the wrong data is introduced.

Testing measures consist of developing a set of test criteria either for the entire system or for specific hardware, software and communications components. For an important and sensitive system such as an electronic voting system, a structured system testing program may be established to ensure that all aspects of the system are thoroughly tested.

Testing measures that could be followed include:

- Applying functional tests to determine whether the test criteria have been met
- Applying qualitative assessments to determine whether the test criteria have been met.
- Conducting tests in “laboratory” conditions and conducting tests in a variety of “real life” conditions.
- Conducting tests over an extended period of time to ensure systems can perform consistently.
- Conducting “load tests”, simulating as close as possible likely conditions while using or exceeding the amounts of data that can be expected to be handled in an actual situation.

Test measures for hardware may include:

- Applying “non-operating” tests to ensure that equipment can stand up to expected levels of physical handling.
- Testing “hard wired” code in hardware (firmware) to ensure its logical correctness and that appropriate standards are followed.

Tests for software components also include:

- Testing all programs to ensure its logical correctness and that appropriate design, development and implementation standards have been followed.
- Conducting “load tests”, simulating as close as possible a variety of “real life” conditions using or exceeding the amounts of data that could be expected in an actual situation.
- Verifying that integrity of data is maintained throughout its required manipulation.

➤ Unit testing

The first test in the development process is the unit test. The sourcecode is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results. Functional and reliability testing in an Engineering environment. Producing tests for the behavior of components (nodes and vertices) of a product to ensure their correct behavior prior to system integration.

➤ Integration testing

Testing is which modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is release.

6. CONCLUSION

In the case of open systems, where the sequences of interactions (stored as system logs) are not constrained by a workflow, process mining techniques whose objective is to extract a process model will usually provide with either overfitting spaghetti models or underfitting flower models, from which little interesting information can be extracted. A more flexible approach is required. In the paper we apply Hashing -based model checking techniques to analyse ecommerce web logs. To enable this analysis, we have proposed a common way of representing event types and attributes considering the e-commerce web structure, the product categorization and the possibilities of users to navigate throw the website according to such organization.

CHAPTER 7

SCOPE FOR FUTURE ENHANCEMENT

The place where tons of online user behavior data are generated every day. These data are used to extract users' valuable information for research purposes or business interests. But, the data are under the risk of being exposed to third parties. Using the data the deceiving advertisements are added with low cost on the same products searched. Here some methods are implemented to perform the data aggregation in a privacy preserving manner. Most of the available methods assure strong privacy protection at the cost of very limited aggregation such as only summation, which hardly satisfies the need of behavior analysis. I propose a model called Privacy Preserving Selective Aggregation (PPSA). This model encrypts the users' sensitive data to prevent privacy from both outside analysts and the aggregation service provider. Also, completely supports selective aggregate functions for online user behavior analysis and guaranteeing differential privacy.

BIBLIOGRAPHY

BOOK REFERENCES

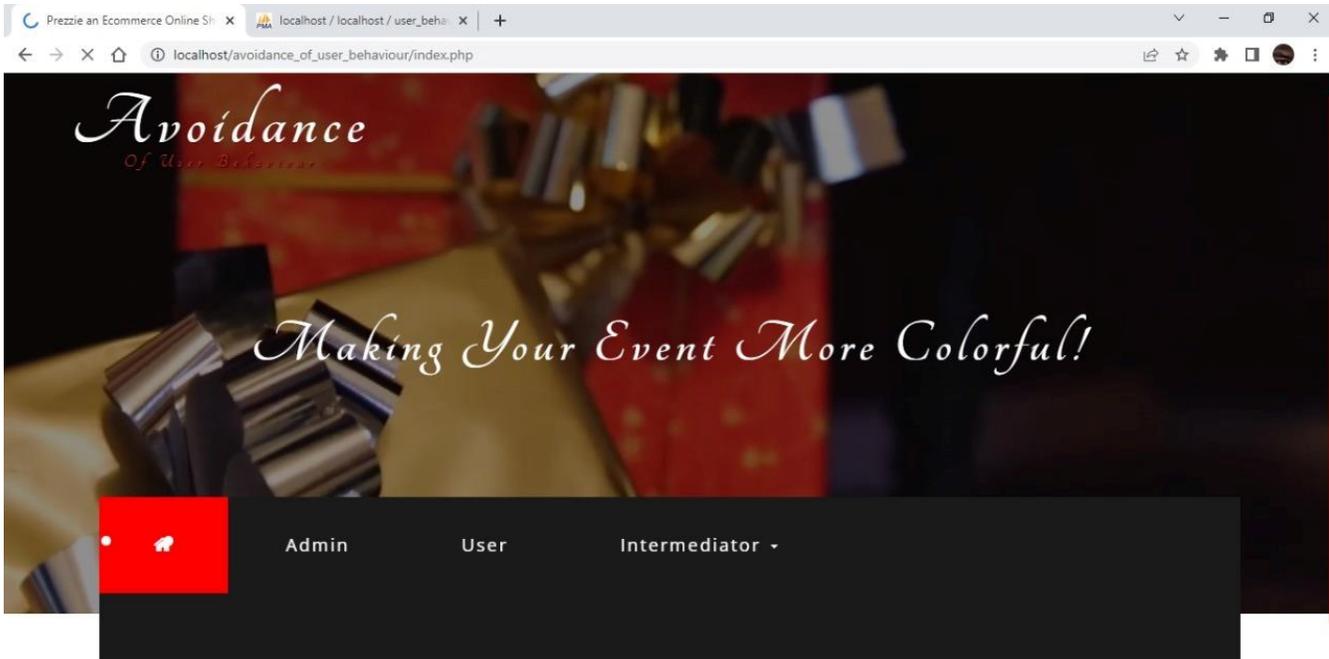
- [1] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the gateway to trust-free cryptographic transactions." in ECIS, 2016, p. 153.
- [2] S. Yin, J. Bao, Y. Zhang, and X. Huang, "M2m security technology of cps based on blockchains," Symmetry, vol. 9, no. 9, p. 193, 2017.

WEB REFERENCES

- [3] <http://ijsrcseit.com/paper/CSEIT184124.pdf>
- [4] https://www.academia.edu/20308192/Online_tender_management_system

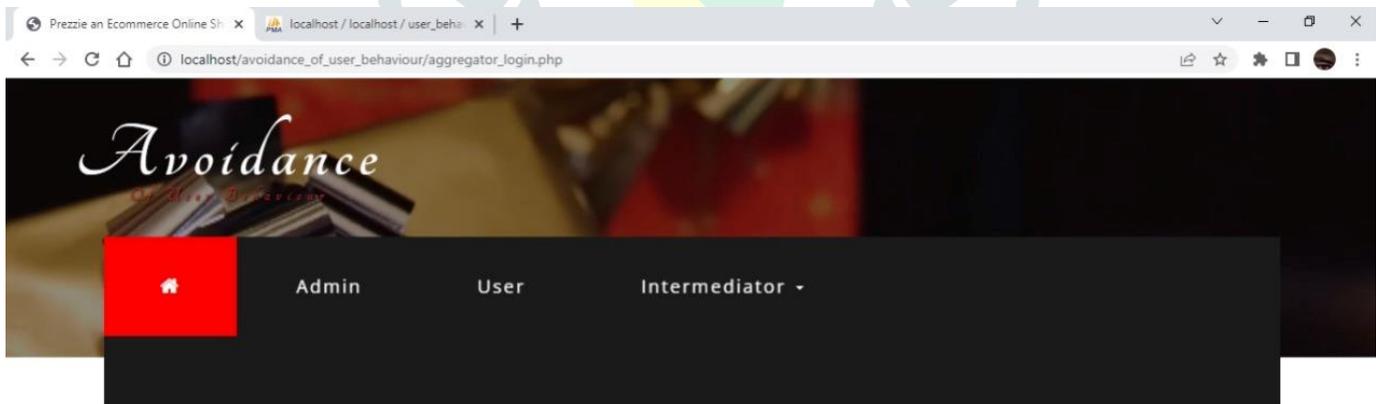


APPENDIX



A.SCREENSHOT

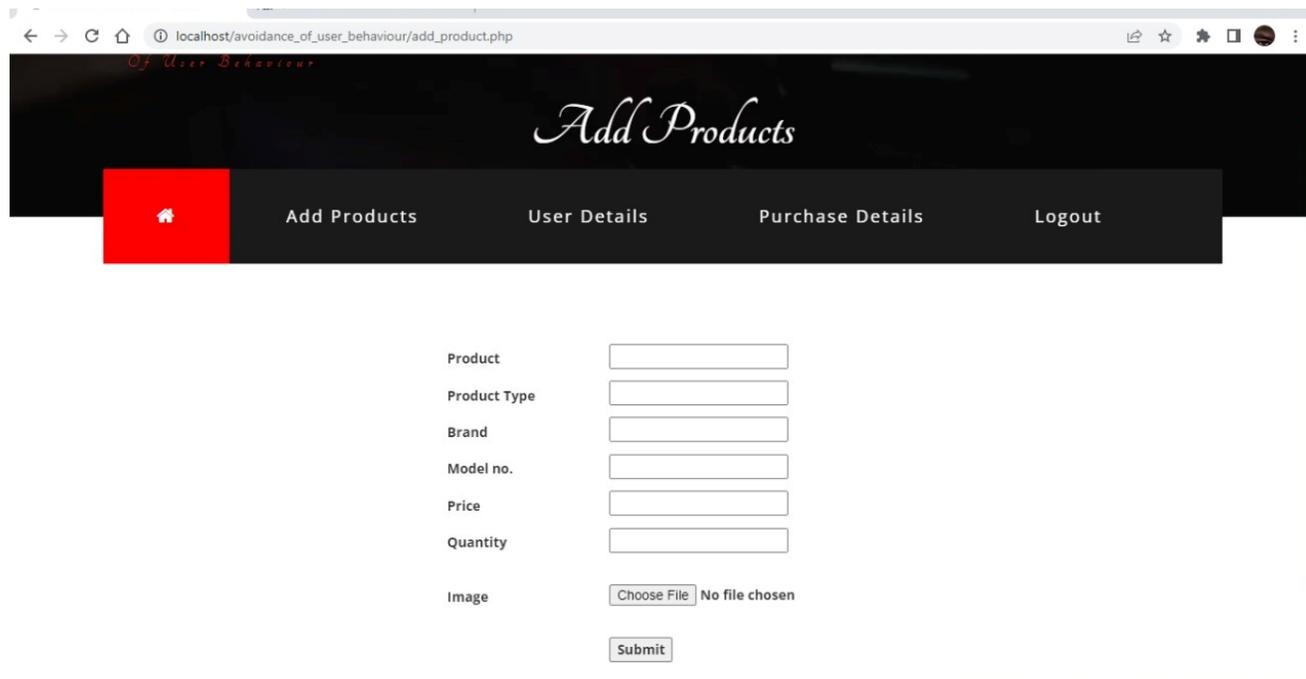
ADMIN HOME



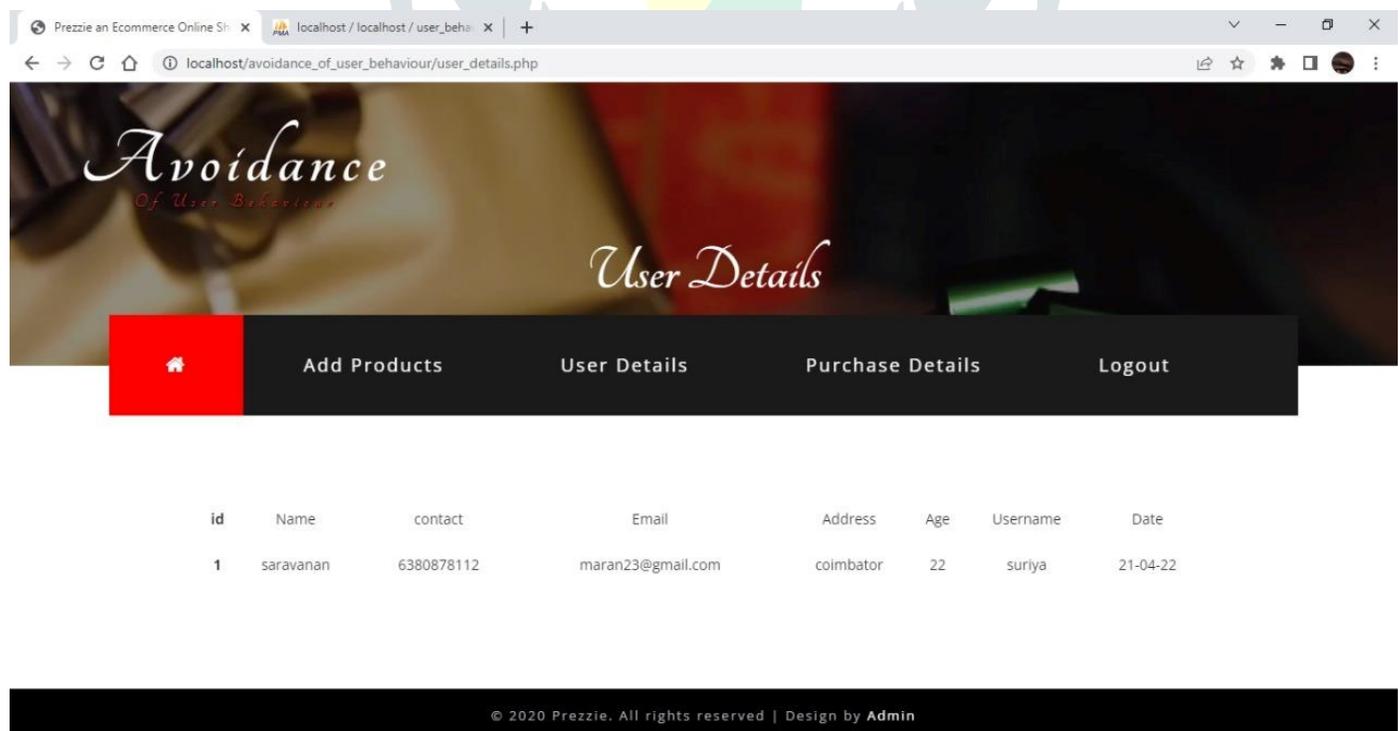
Username

Password

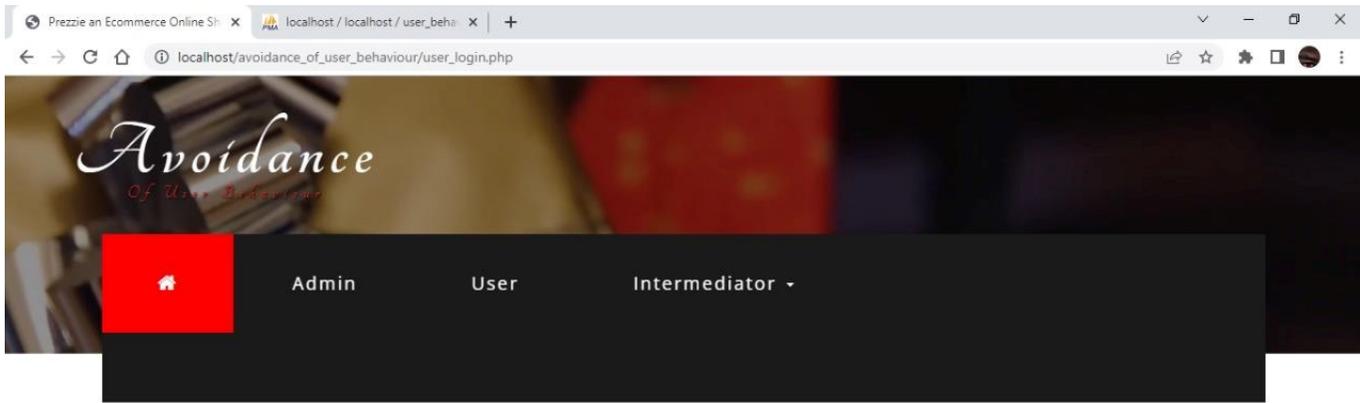
ADD PRODUCTS



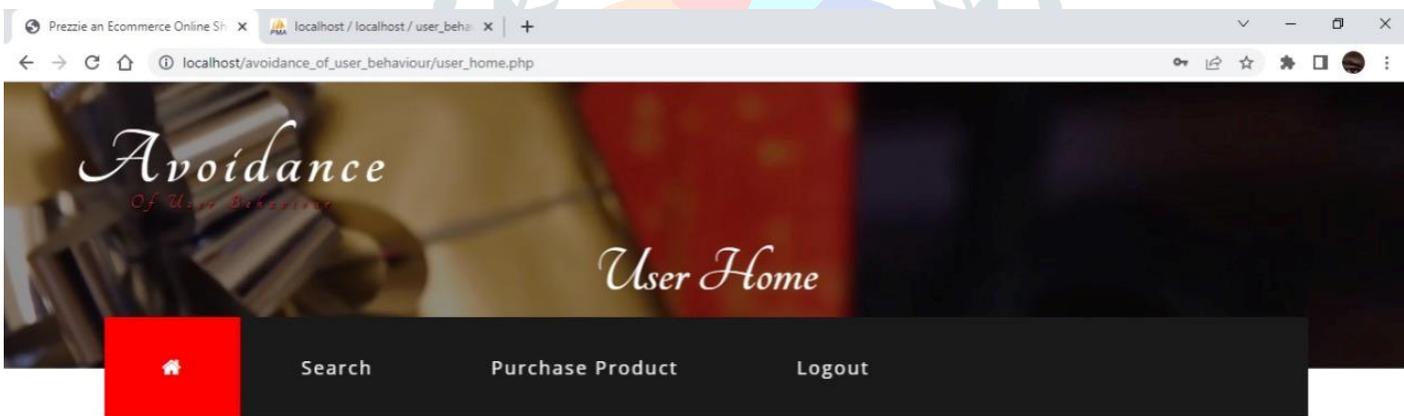
USERS DETAILS



USER LOGIN



USER HOME



Welcome : suriya

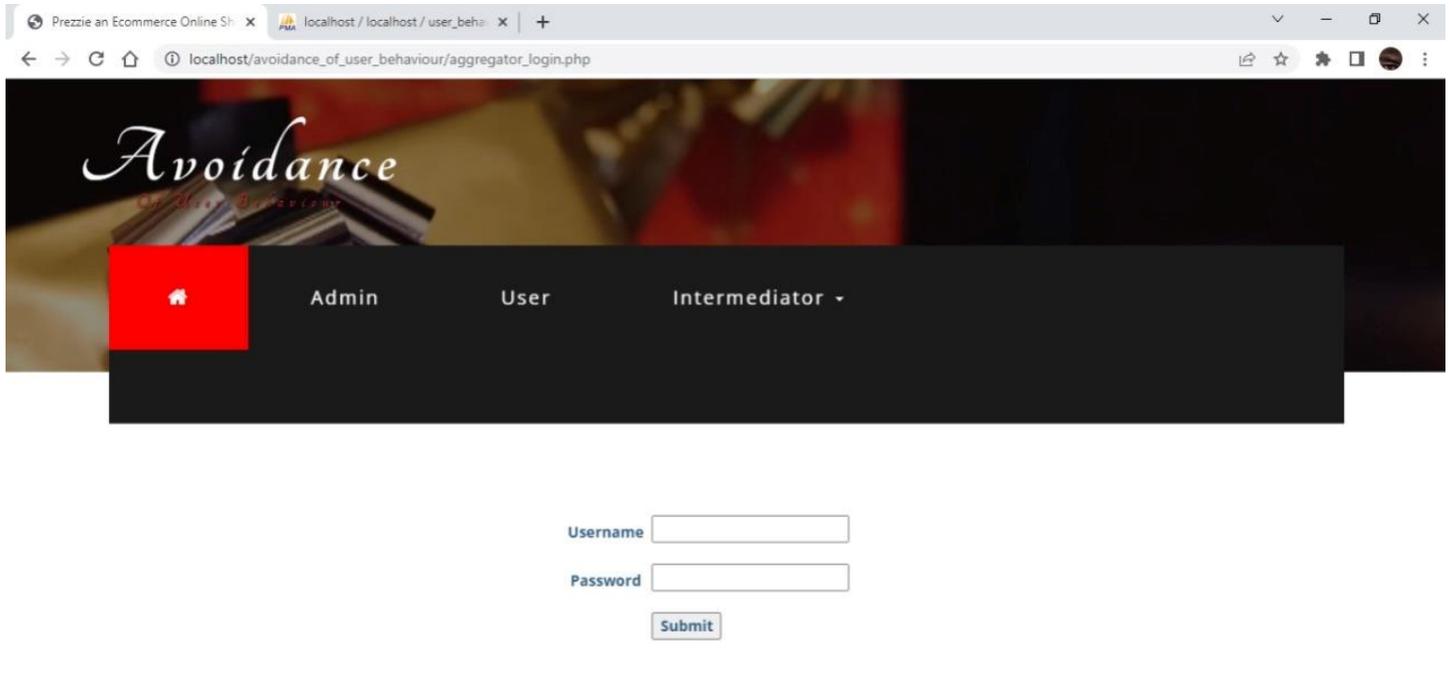
USER SEARCH PRODECT

| id | product | Product type | Brand | Model | Price | Quantity | Image | Buy Now |
|----|---------|--------------|-------|------------|-------|----------|-------|----------------------------|
| 1 | mobile | electronic | redmi | note 8 | 13000 | 92 | | Click Here |
| 2 | watch | electronic | TaTa | A1 | 1500 | 41 | | Click Here |
| 3 | shirt | sdsd | TaTa | sfsdf | 1000 | 47 | | Click Here |
| 4 | saree | saree | saree | 1521 | 250 | 497 | | Click Here |
| 5 | asus | laptop | asus | 01 | 65000 | 27 | | Click Here |
| 6 | system | laptop | DELL | 9843566721 | 85000 | 4 | | Click Here |

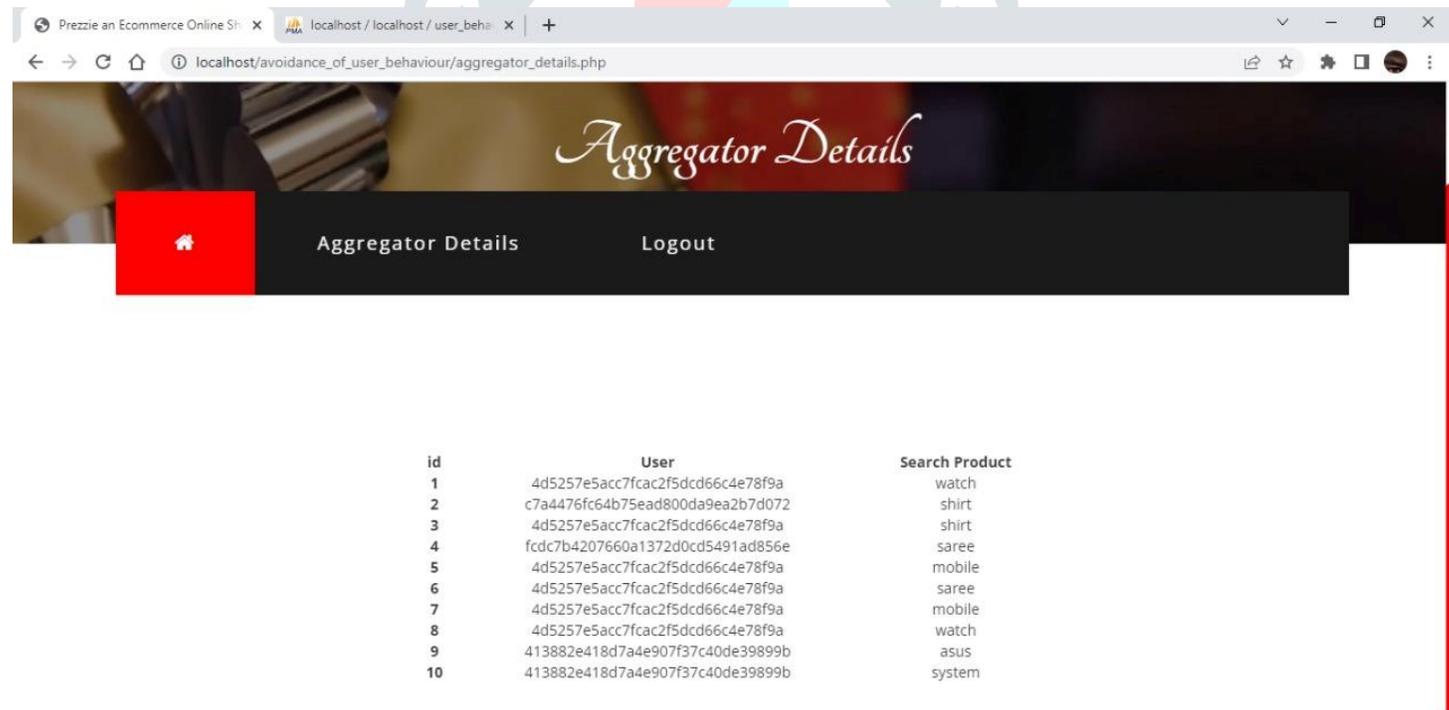
PURCHASE PRODUCTS

| id | Uname | product | Product type | Model | Price | Quantity | Total Amount | Image | Date |
|----|--------|---------|--------------|------------|-------|----------|--------------|-------|------------|
| 17 | suriya | asus | laptop | 01 | 65000 | 27 | 65000 | | 21-04-2022 |
| 18 | suriya | system | laptop | 9843566721 | 85000 | 4 | 85000 | | 12-05-2022 |

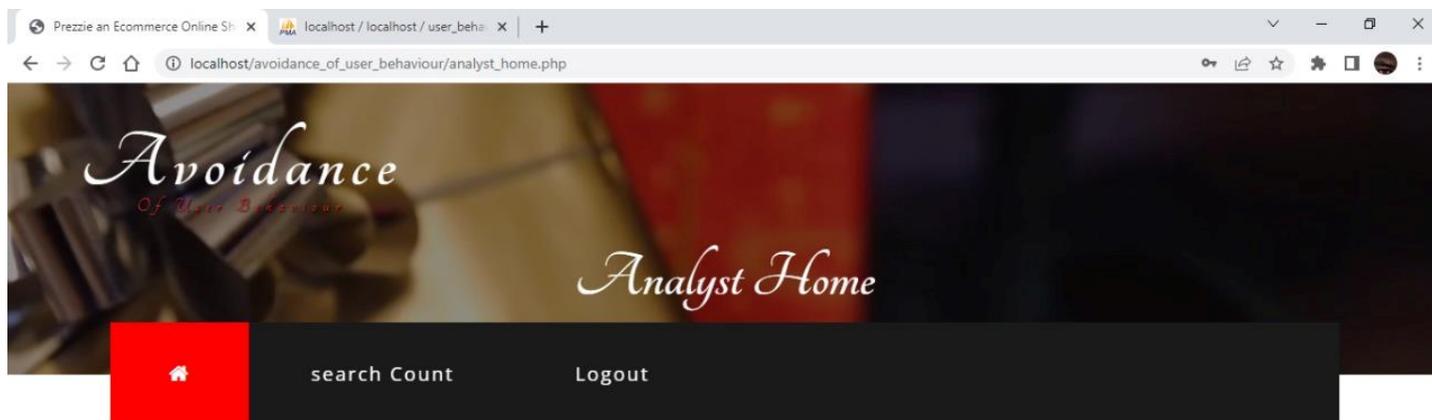
AGGREGAT LOGIN



AGGREGATOR DETAILS



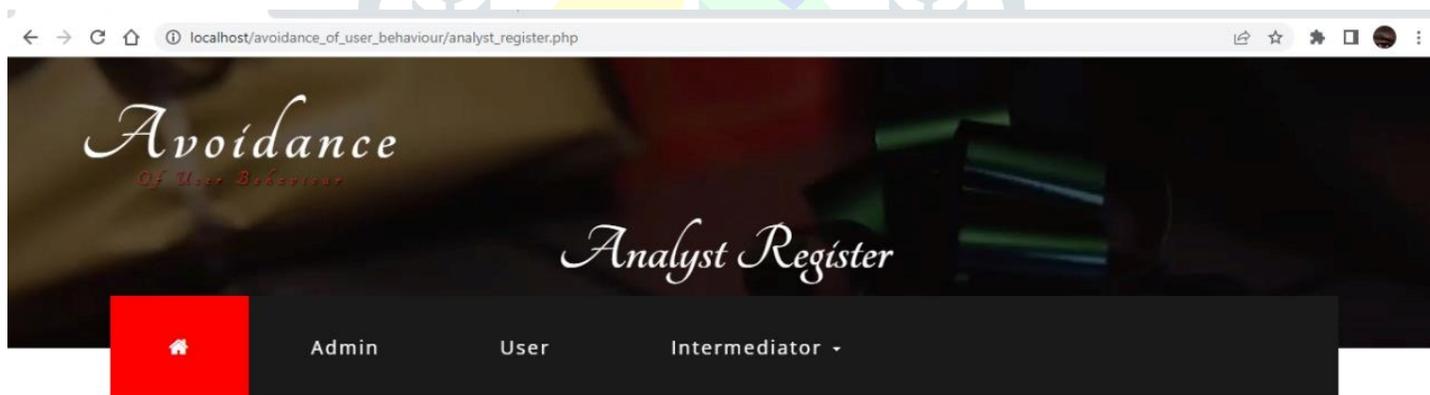
ANALYST HOME



Analyst Home

© 2020 All rights reserved | Design by Admin

ANALYST REGISTER

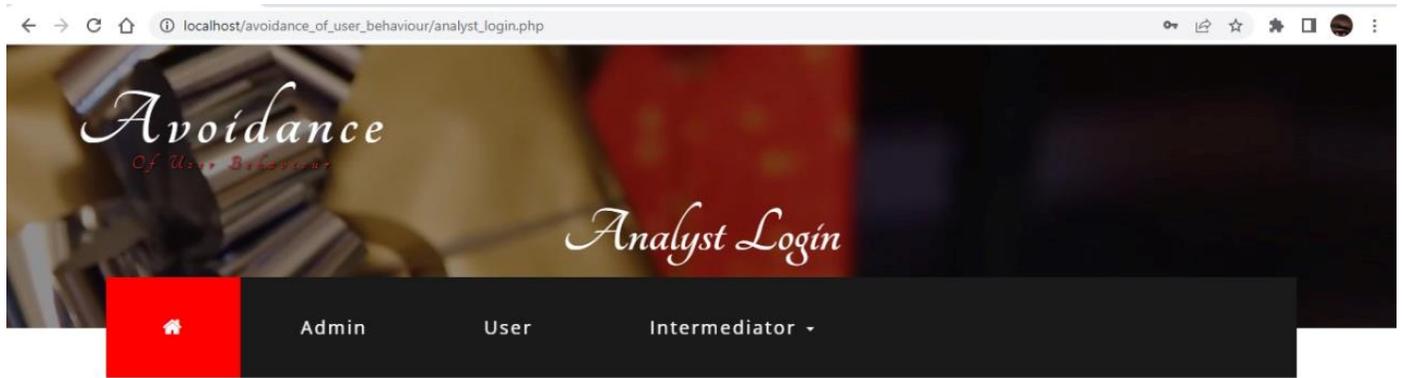


Name

contact

Email

ANALYST LOGIN

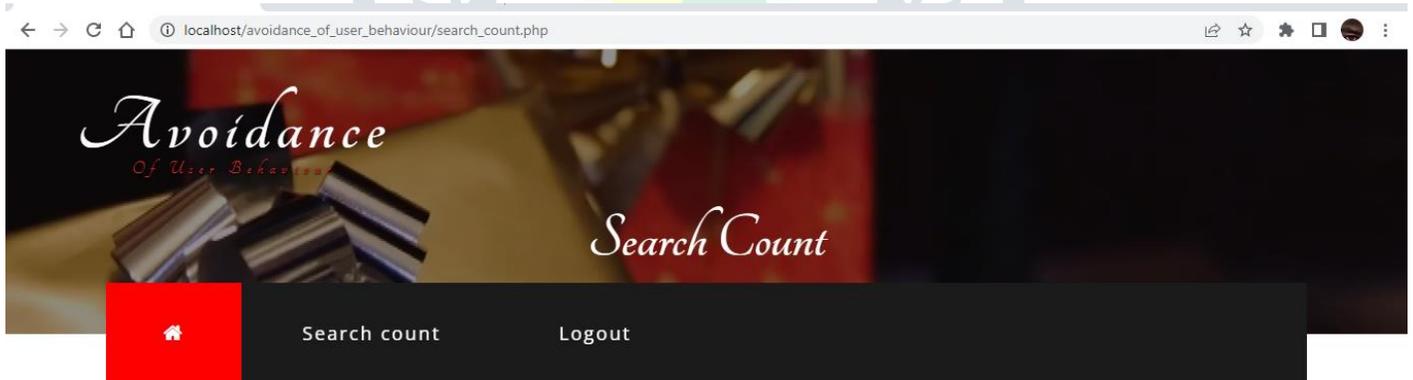


OTP

Submit

REGISTRATION

SEARCH COUNT



| Id | Search Value | Count |
|----|--------------|-------|
| 1 | mobile | 17 |
| 2 | watch | 17 |
| 3 | shirt | 1 |
| 4 | saree | 3 |
| 5 | asus | 2 |
| 6 | system | 0 |