**1.1 OVERVIEW**

**The Internet of Things (IoT)** is the network of physical objects accessed through the Internet that can identify themselves to other devices and use embedded technology to interact with internal states or external conditions.

When we talk about interconnection, we usually think in terms of computers, tablets and smart phones. IoT describes a world where just about anything can be connected and communicate in a “smart mode” by combining simple data to produce usable intelligence. With the IoT, the physical world is becoming one big information system with an ultimate goal of improving quality of life and empowering new business models.

### Why Security is important in IoT?

Security in IoT is important due to following,

* Ineffective authentication
* Lack of transport encryption
* Insecure web interface

#### Ineffective authentication

#### Securing IoT devices is challenging for several reasons. A rapidly increasing number of gadgets are becoming smart devices, and as manufacturers roll out new products more quickly, security can be given low priority as the focus is on time-to-market and return-on-investment metrics. A lack of awareness among consumers and businesses is also a major obstacle to security, with the convenience and cost-saving benefits of IoT tech appearing to outweigh the potential risks of data breaches or device hacking. An unsecure connected toy, though, poses an entirely different sort of danger than hacking into a computer. Malicious hackers could use these toys to gain access to the home’s internet or communicate with and even physically harm children.

#### Lack of transport encryption

If your device is sending private information over an insecure protocol anyone could be reading it. And it’s not always obvious to people what information an IoT device might be sharing.

#### Insecure web Interface

Cyber-intruders don’t always need complex methods to compromise devices on your home network. Sometimes, reaching critical information is just a matter of asking for it in the right way, which can basically be an SQL injection. SQL (Structured Query Language) is just a way to handle entries in a database, which can be accessed through a web interface. A hacker wants to tap into databases that hold sensitive data like usernames, passwords and permissions, and alter or use them to gain a foothold in your virtual perimeter. This, in turn, could bring them closer to accessing other connected devices and, ultimately, taking over the entire network. An SQL injection attack works when the hacker imitates a user who is allowed to execute a limited set of requests: the attacker inputs a valid request and mixes in new instructions that also get executed. This could be compared to a robot tasked to manage goods in a sealed room. Although it can do more, the robot’s purpose is to execute a specific set of actions, such as moving boxes, getting them out and in or showing the record of the storage room. However, if a user injects a request to smash things up, the robot won’t know the difference and will obey the command. The risks associated with this type of attack are serious.

All of the above reasons show that there is a lot of demand for **IoT** **security.**

The aim of Evaluation System for IoT is to provide the solution for all the above mentioned vulnerabilities.

**1.2 STATEMENT OF THE PROBLEM**

Design and develop the application tool for solving the problem of ineffective authentication, lack of transport encryption, insecure web interface in IoT devices.

**1.3 MOTIVATION**

As IoT grows, the attack surface also grows and all the loopholes/vulnerabilities present in the digital world will flow into our real world. Before IoT, attackers used vulnerabilities for data theft or to make money or sometimes just for fun, but with IoT, the attack surface has grown to such extent that attacker can use vulnerabilities or loopholes in the car, smart sniper rifle etc., to kill a person remotely with a few strokes of the keyboard. Attackers are constantly finding the vulnerabilities to break into IoT and use those vulnerabilities for many illegal purposes. The exponential rise in the use of IoT devices, more processing of sensitive data by these devices, and their mass exploitation is the motivation for the development of this evaluation system.

**1.4 MERITS**

* This system can be used by the users of IoT devices like smart bulbs, routers, and so on.
* The system will secure your data from SQL injection to the device.
* It provides an effective authentication mechanism to secure your device from attack.
* Encrypts your data while transferring it over the channel

**1.5 ORGANISATION OF REPORT**

This report contains the project related to the security of IoT devices which will provide the security at three levels, security at device level, data transmission and storage. Here the main intension is to secure the data of IoT users. The report contains several chapters.

The first chapter contains the introduction to the topic. Followed by, it contains overview of the topic. Next, we have problem statement which will say the main intension of the project. In the next section, we have motivation of the project which explains the main reason for choosing this topic. Next, we have merits of proposed system.

The second chapter is literature survey. It contains the topics that are referred from several papers.

The third chapter explains about the software requirement specification. That is software and hardware used for the project.

Chapter four explains about the system design. This contains architecture of the proposed system, overview, database design, and so on.

Chapter five explains the implementation of the project. Here modules used in the system are explained.

Chapter six is the testing. That tests the project and results are displayed in that chapter. That means test cases are included in this chapter.

At the last, in chapter seven, we include conclusion and future works of the project. Here have I explained how the system can be modified in future for better output.

**LITERATURE SURVEY**

Omner Barajas in [1] says that, Privacy is a serious concern not just in the IoT, but in all the applications, devices or systems where we share information. Even when users take precautions to secure their information, there are conditions that are beyond their control. Hackers can now craft attacks with unprecedented sophistication and correlate information not just from public networks, but also from different private sources, such as cars, smartphones, home automation systems and even refrigerators. Currently, more things are connected to the Internet than people, according to an infographic from Cisco. It goes on to say that **25 billion devices are expected to be connected by 2015 and 50 billion are slated to connect by 2020**. In this quickly evolving world, all the things that connect to the Internet are exponentially expanding the attack surface for hackers and enemies. A recent study showed that **70 percent of IoT devices contain serious vulnerabilities**. There is undeniable evidence that our dependence on interconnected technology is defeating our ability to secure it. The industry needs to learn from its mistakes as it innovates and builds devices to function interconnectedly with the Internet. Many of the best security practices can be leveraged, such as hardening the systems, using secure protocols for communication or installing the latest updates, fixes and patches. Innovators need to consider that future security will be managed automatically by the system instead of users, and designing secure technology will require a new approach and mindset. Information technology security experts have been warning the public about cyber threats for years, using conferences such as Black Hat to publicize new vulnerabilities in systems and software. At the **2014 Black Hat conference**, many discussions were focused on the IoT. Still, users seem not to pay attention to these alerts either because they aren’t particularly knowledgeable on the technical aspects of exploits, vulnerabilities and threats, or simply because they do not care.

Zlata Rodionova in [2] reported that, TalkTalk has been fined a record £400,000 fine for security failings which led to the theft of personal data of almost 157,000 customers. The cyber attack in October last year exposed the latest security failure for the company, which was forced to admit it had not encrypted some personal details of customers. The Information Commissioner's Office (ICO) said the attack could have been prevented if TalkTalk had taken basic steps to protect customers’ information. Almost 157,000 customers had their details stolen, including bank account numbers, birth dates and addresses. Elizabeth Denham, the Information Commissioner, said: “TalkTalk’s failure to implement the most basic cyber security measures allowed hackers to penetrate TalkTalk’s systems with ease“. An investigation by the ICO found hackers gained access to the database of details which TalkTalk had from its takeover of rival firm Tiscali via vulnerable web pages which it had not spotted.

Alfred NG in [3],has told, This past summer, security researchers from IBM and Threatcare identified 17 vulnerabilities from four smart cities built on leading smart city systems deployed across the world. Out of the 17 zero-day bugs, eight were found to be critical. Many of the vulnerabilities were caused by elementary flaws in security design, such as allowing the use of default passwords and leaving networks unsecured online, making these systems accessible even for amateur hackers. Particularly concerning were the discoveries of authentication flaws and encryption issues in server communications systems, as both technologies are essential for preventing security breaches. With the number of IoT system vendors growing in a regulation-wise lax environment, it’s important for vendors and users alike to hold each other responsible and challenge their standards of security–especially when it comes to critical infrastructure.

[4] If you’ve kept an eye on recent developments in IoT, chances are you have heard about vulnerable healthcare devices. In January 2017, two renown security researchers disclosed severe vulnerabilities in the Medtronic CareLink 2090, a monitoring device that doctors use to control pacemaker settings. Again, poor authentication and encryption features left the device software vulnerable to malware infections. When the researchers shared updates on the case at the 2018 Black Hat conference last August, many were shocked to hear that some of the vulnerabilities still persisted. This was despite notifying Medtronic of the security flaws 570 days ago and delivering a proof-of-concept 155 days ago, as of August 9.In most cases, vendors tend to step up when vulnerabilities are disclosed to the public, even if they did not actively respond to any research shared in private. Yet as this case shows, vulnerabilities can remain unaddressed for extended periods of time, even when dealing with something as sensitive as medical devices that could be used to threaten patients’ lives.

**3.1 SOFTWARE REQUIREMENTS**

|  |  |
| --- | --- |
| OPERATING SYSTEM | Windows 7 Professional |
| LANGUAGE USED | Java Hibernate |
| DATABASE | MySQL |
| IDE | NetBeans 8.2 |

**Table 3.1 Software requirements**

**3.2 HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| SYSTEM | Intel /Pentium |
| HARDDISK | 2 GB+ |
| RAM | 250 GB+ |

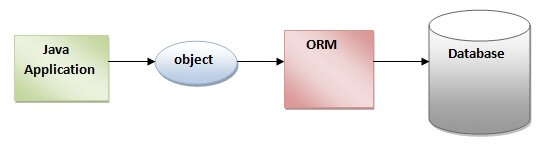
**Table 3.2 Hardware requirements**

**3.3 SOFTWARE ENVIRONMENT**

**3.3.1 Hibernate Framework:**

Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.



**Fig.3.1 Hibernate framework**

The ORM tool internally uses the JDBC API to interact with the database.

**3.3.2 Advantages of Hibernate Framework:**

Following are the advantages of hibernate framework:

### 1) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

### 2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.

### 3) Database Independent Query

HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.

### 4) Automatic Table Creation

Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

### 5) Simplifies Complex Join

Fetching data from multiple tables is easy in hibernate framework.

### 6) Provides Query Statistics and Database Status

Hibernate supports Query cache and provide statistics about query and database status.

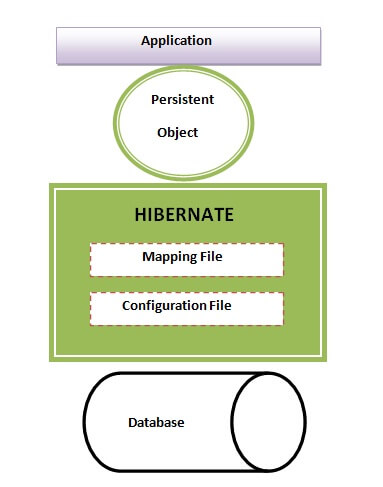
**3.3.3 Hibernate Architecture**

The Hibernate architecture includes many objects such as persistent object, session factory, transaction factory, connection factory, session, transaction etc.

The Hibernate architecture is categorized in four layers.

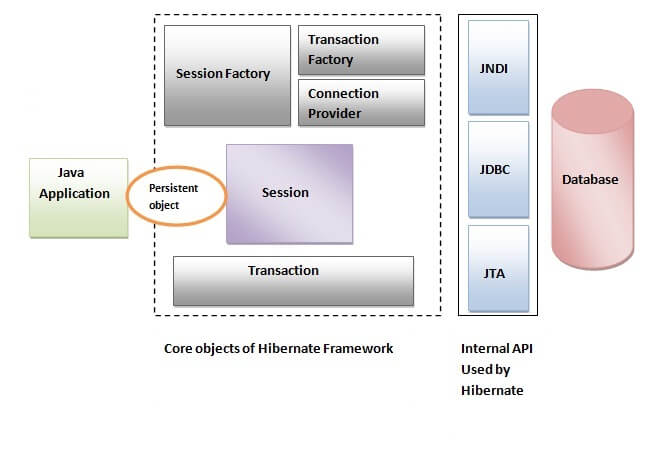
* Java application layer
* Hibernate framework layer
* Backhand API layer
* Database layer

Let's see the diagram of hibernate architecture:



**Fig.3.2(a) Hibernate architecture**

This is the high level architecture of Hibernate with mapping file and configuration file.



**Fig.3.2(b) Hibernate architecture**

Hibernate framework uses many objects such as session factory, session, transaction etc. along with existing Java API such as JDBC (Java Database Connectivity), JTA (Java Transaction API) and JNDI (Java Naming Directory Interface).

**4.1 OVERVIEW**

IoT devices are much in use today. The manufacturer will have much interest in the profit obtained by selling the product. He never bothers about the security aspect and provides simple passwords like admin or admin123 for all the devices. But this makes trouble for the customers. The malicious users can easily access to such devices.Similarly, the data can be accessed when it is transmitting over the channel. Also, the data stored in database can be attacked by simple methods like SQL injection.

By keeping all these things in mind, I propose a system called Evaluation system for IoT which will provide solutions for these problems.

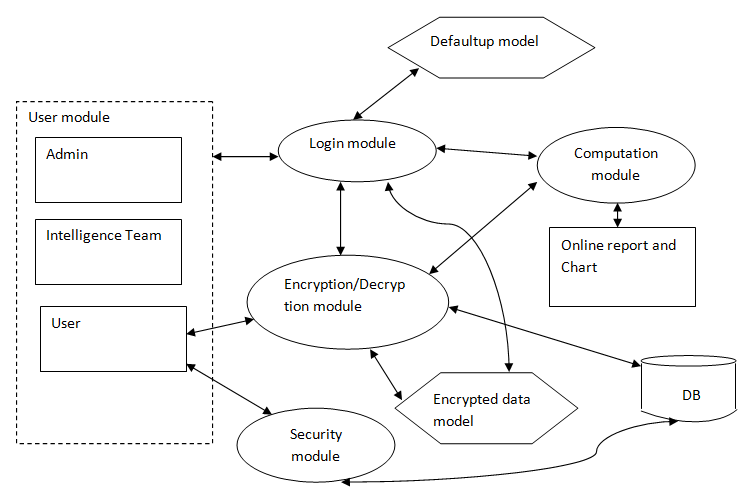
**4.2 ARCHITECTURE OF PROPOSED SYSTEM**

**4.2.1 INTRODUCTION**

System design is the process or art of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and system engineering.

In system design focus is on deciding which modules are needed for the system, the specifications of these modules should be interconnected is called system design.

System design also called top-level design. Here we consider a system to be set of components with clearly defined behaviour that interact with each other in a fixed manner to produce some behaviour. In a system design, the design consists of module definitions, with each module supporting a functional abstraction.





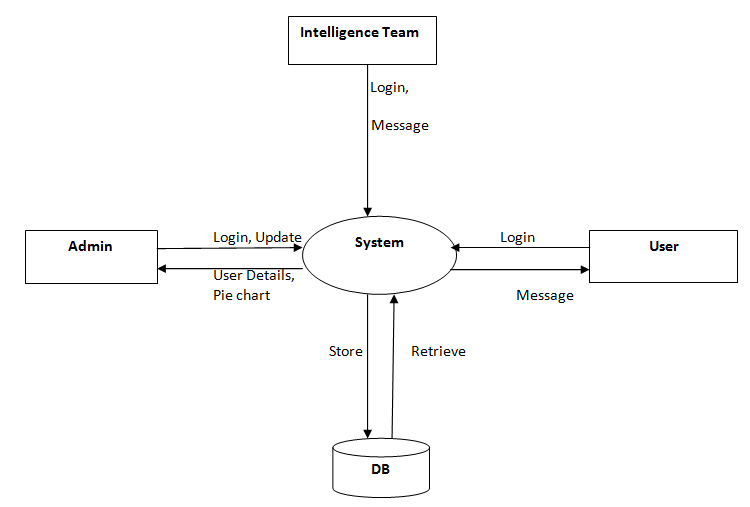
**Fig.4.1 Architecture of proposed system**

**4.2.2 DESCRIPTION OF PROGRAMS**

**4.2.2.1 CONTEXT FLOW DIAGRAM**

A context flow diagram (CFD) is a graphical representation of the “flow” of context through an information system.

CFD can also be used for the visualization of data processing (structured design). On a CFD, data items flow from an external data source or an internal data source to an internal data store or an external data sink, via an internal process. A CFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the control through an algorithm, allowing a reader to determine what operations will be performed, in that order, and what circumstances.

****

**Figure 4.2 Context flow diagram (CFD)**

**4.2.2.2 DATA FLOW DIAGRAM**

A data flow diagram is a graphical representation of the “flow” of data through an information system, modelling 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).

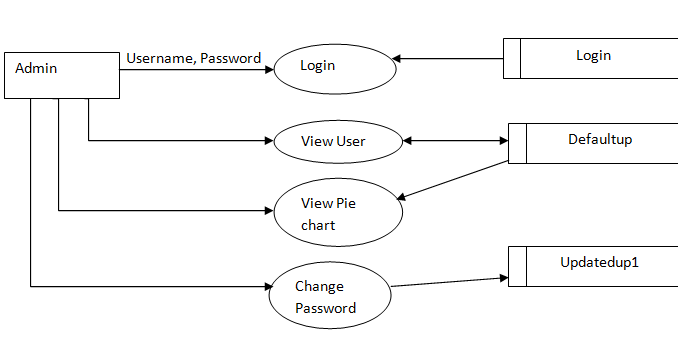
The DFD also called bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

**Some of the symbols used in data flow diagrams are given below:**

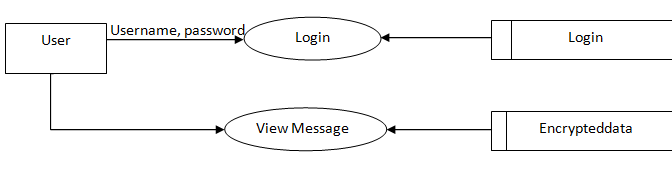
|  |  |
| --- | --- |
| Process | An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD. |
| External entity | Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed. These data flows are the inputs and outputs of the DFD. |
| Data flow | A data flow shows flow of information from source to destination. A data flow is represented by a line, with arrowhead showing the direction of flow. |
| Data store | A data store indicates to which database a particular data is stored and from which database the data is retrieved. |

**Table 4.1 Symbols used in data flow diagram (DFD)**

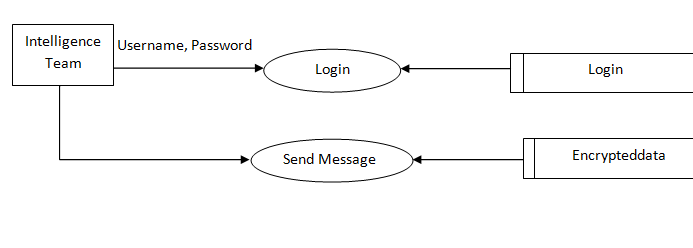
**DATAFLOW DIAGRAM:**

****

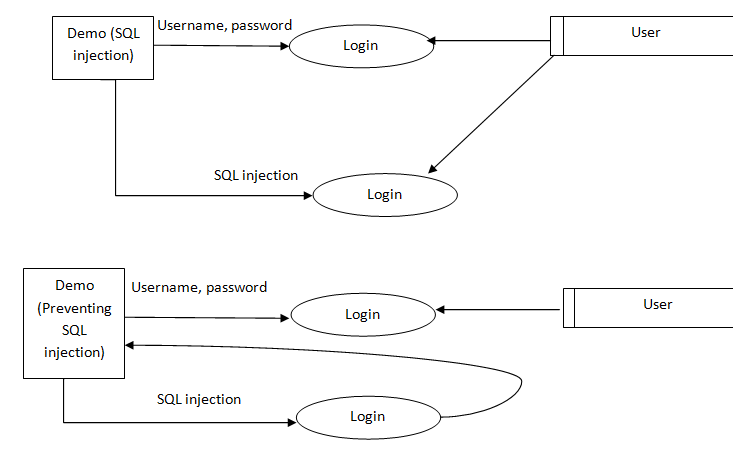
**Fig: 4.3 Data Flow Diagram for Admin**

****

**Fig: 4.4 Data Flow Diagram For User**

****

**Fig.4.5 Data Flow Diagram for Intelligence Team**

****

**Fig.4.6 Data Flow Diagram for SQL injection Demo**

**4.3 DATABASE DESIGN**

|  |  |  |
| --- | --- | --- |
| **Field\_name** | **Data type** | **Description** |
| SerialNo | Int(11) | PrimaryKey, NotNull |
| UserName | Varchar(50) | NotNull |
| Password | Varchar(20) | NotNull |
| Item | Varchar(50) | NotNull |
| PortNo | Varchar(25) | NotNull |
| Brand | Varchar(50) | NotNull |

**Table 4.2 Defaultup**

|  |  |  |
| --- | --- | --- |
| **Field\_name** | **Data type** | **Description** |
| serialno | Int(11) | PrimaryKey, NotNull,  Auto increment |
| message | Varchar(45) | NotNull |
| reciever | Varchar(50) | NotNull |
| time | Timestamp | NotNull |
| keyVal | Varchar(50) | NotNull |

**Table 4.3 Encrypteddata**

|  |  |  |
| --- | --- | --- |
| **Field\_name** | **Data type** | **Description** |
| serialno | Int(11) | PrimaryKey, NotNull,  Auto increment |
| id | Varchar(50) | NotNull |
| password | Varchar(45) | NotNull |

**Table 4.4 Login**

|  |  |  |
| --- | --- | --- |
| **Field\_name** | **Data type** | **Description** |
| SerialNo | Int(11) | PrimaryKey, NotNull |
| UserName | Varchar(500) | NotNull |
| Password | Varchar(500) | NotNull |
| Item | Varchar(50) | NotNull |
| PortNo | Varchar(50) | NotNull |
| Brand | Varchar(50) | NotNull |

**Table 4.5 Updatedup1**

|  |  |  |
| --- | --- | --- |
| **Field\_name** | **Data type** | **Description** |
| id | Varchar(12) | PrimaryKey,  NotNull |
| password | Varchar(12) | NotNull |
| deviceinfo | Varchar(12) | NotNull |

**Table 4.6 User**

**4.4 E-R DIAGRAM**

A data model is a conceptual representation of the data structures that are required by a database. The first step in designing a database is to develop an Entity Relationship Diagrams (ERD).

The ERD serves as a blue print from which a relational database may be deduced. Here, when Entity A matches exactly one record in Entity B and every record in B matches exactly one record in entity A it is called one-to-one mapping. One-to- many means that every record in A matches zero or more records in B and every record in B matches exactly one record in A. If there is a one-to-many relationship between two entities, then these entities are represented as Associative Entities.

Entity Relationship analysis uses three major abstractions to describe data. These are entities which are distinct things in the enterprise. Relationships are meaningful interactions between the objects and the attributes which are the properties of entities and relationship. We can express the overall logical structure of database graphically with an E-R diagram.

The symbols used in E-R diagram are:-

* **Entity**:

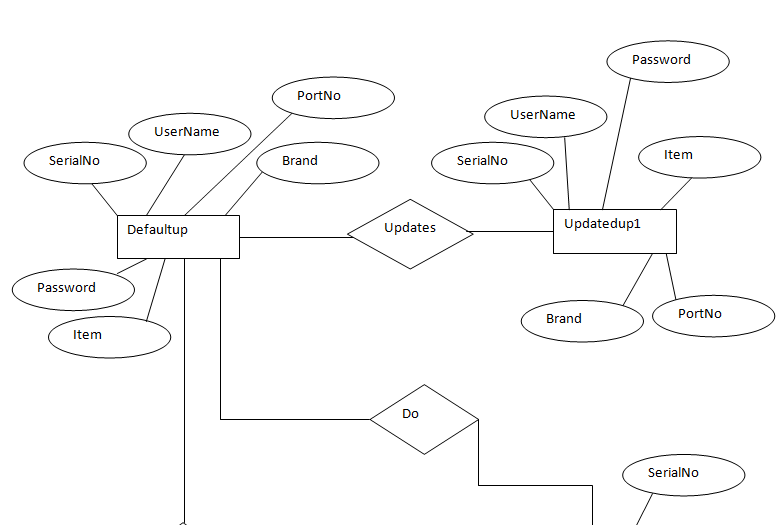
Entity is a “thing” in the real world with an independent existence. An entity may be an object with a physical existence such as person, car or employee. Entity symbol is as follows:

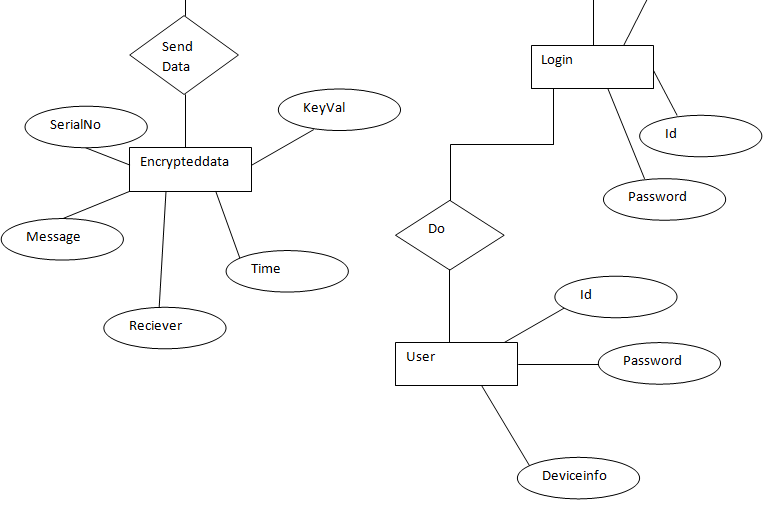
* **Attribute:**

Attribute is a particular property that describes the entity. Attribute symbol is:

* **Relationship:**

Relationship will be several implicit relationships among various entity types whenever an attribute of one entity refers to another entity type some relationship exits. Relationship symbol is:

****

****

**Figure 4.7 Entity relationship model of the proposed system**

**5.1 MODULES**

This project has 7 modules.

1. Login module:

This module is used to login to the system by admin, user intelligence team. We can login to the proposed system by giving user name and password.

1. Display module:

This module will display the details of IoT device users which is stored in the database. Only admin can login and see this list.

1. Report generation module:

This module will analyse the number of users having same username and password from the data being stored in the database. Only admin has the access to this module.

1. Update Password module:

This module will change the password of the device based on the selection done by admin. It will generate a random encrypted password using BASE64Encoder algorithm.

1. Encryption module:

In this module, the sensor will send the data to the server. In order to protect the data from malicious access we are encrypting it and sending it to the server.

1. Decryption module:

In this module, the message or the data that is being sent from the server is decrypted and showed to the server. The server can access the data only when it login.

1. Security module:

This module shows a demo of SQL injection and how to prevent SQL injection using prepared statements. The vulnerable page will help you achieve SQL injection where as non vulnerable page will not.

**5.2 Data Encryption Standards (DES)**

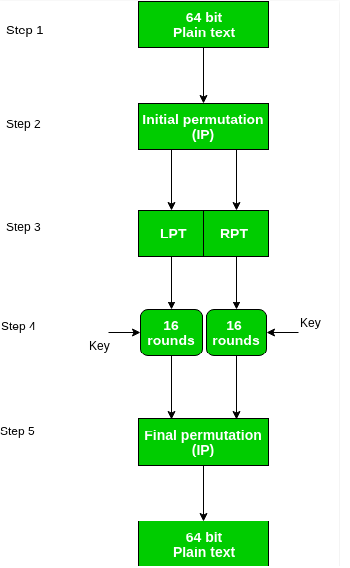
The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).DES is a block cipher, and encrypts data in blocks of size of 64 bit each, means 64 bits of plain text goes as the input to DES, which produces 64 bits of cipher text. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits.

We have mention that DES uses a 56 bit key. Actually, the initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56 bit key. That is bit position 8, 16, 24, 32, 40, 48, 56 and 64 are discarded.

Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key.

DES is based on the two fundamental attributes of cryptography: substitution (also called as confusion) and transposition (also called as diffusion). DES consists of 16 steps, each of which is called as a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

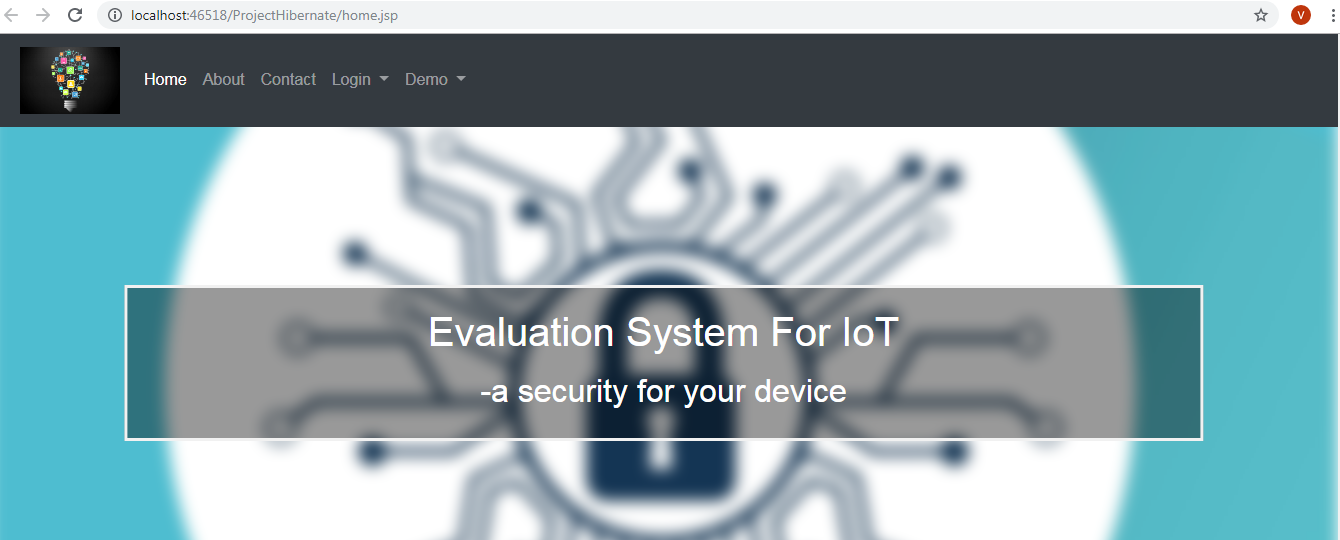
1. In the first step, the 64 bit plain text block is handed over to an initial Permutation (IP) function.
2. The initial permutation performed on plain text.
3. Next the initial permutation (IP) produces two halves of the permuted block; says Left Plain Text (LPT) and Right Plain Text (RPT).
4. Now each LPT and RPT to go through 16 rounds of encryption process.
5. In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
6. The result of this process produces 64 bit cipher text.



**Fig.5.1 Working of DES algorithm**

* 1. **SCREENSHOTS**

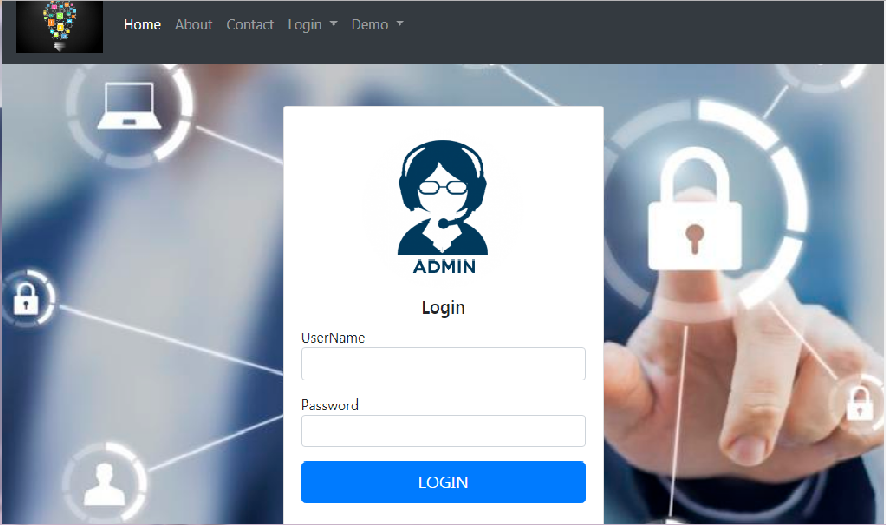
1. **Home page**

****

**Fig.5.2 Home page**

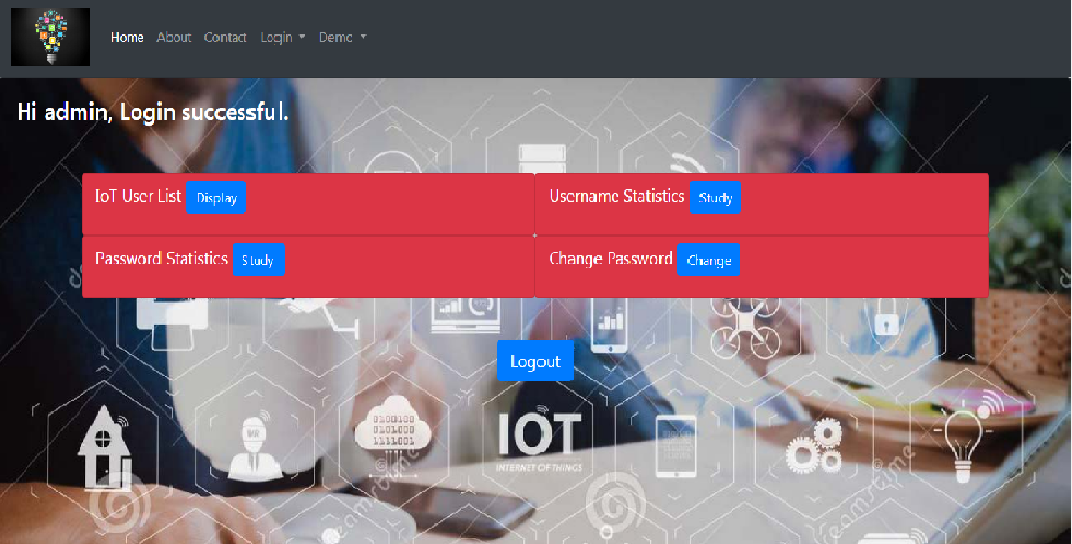
This is the home page where you can navigate to other remaining pages. We can use Login dropdown list present in the navigation bar to choose the type of users being logged in. The users may be admin, intelligence team or user.

1. **Admin page**



**Fig.5.3 Admin page**

Hereadmin can login to the system. Here we have to enter the username and password and select the login button. If the password is wrong, then the page will be redirected to the Home page.



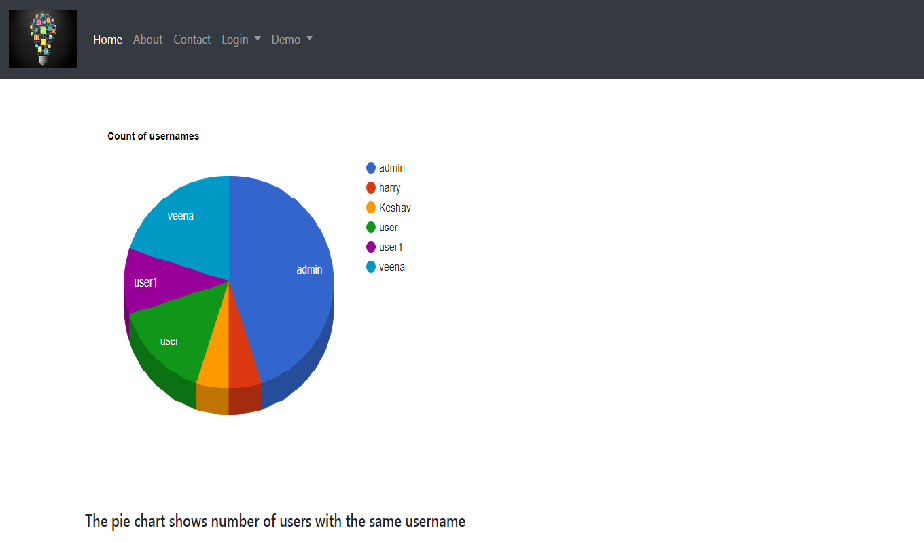
**Fig.5.4 Admin login successful**

When the login is successful, above page will open. Here, the admin can see the list of IoT device users, analyse the username and password by looking into the pie chart that is being generated, and also, admin can change the password if he finds it weak.



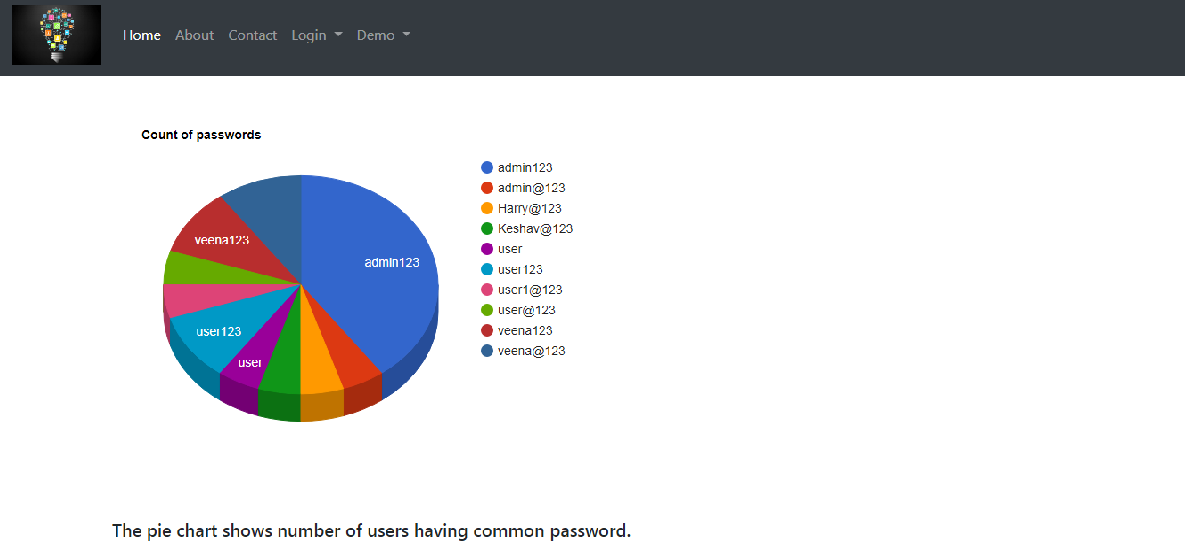
**Fig.5.5 List of IoT device**

The above list shows the details of IoT device users.



**Fig.5.6 Count of username**

The pie chart shows the number of users having same username. By analysing the chart, admin can understand which username has more number of counts.



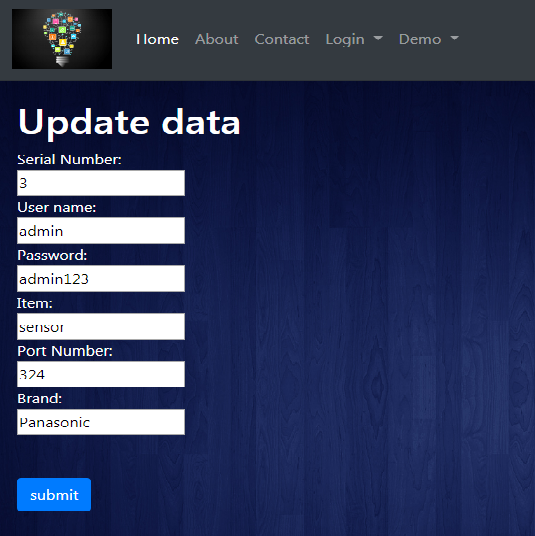
**Fig.5.7 Count of password**

The above chart shows the number of users having same password, which will obviously a vulnerable case. The admin can analyse and change the password.

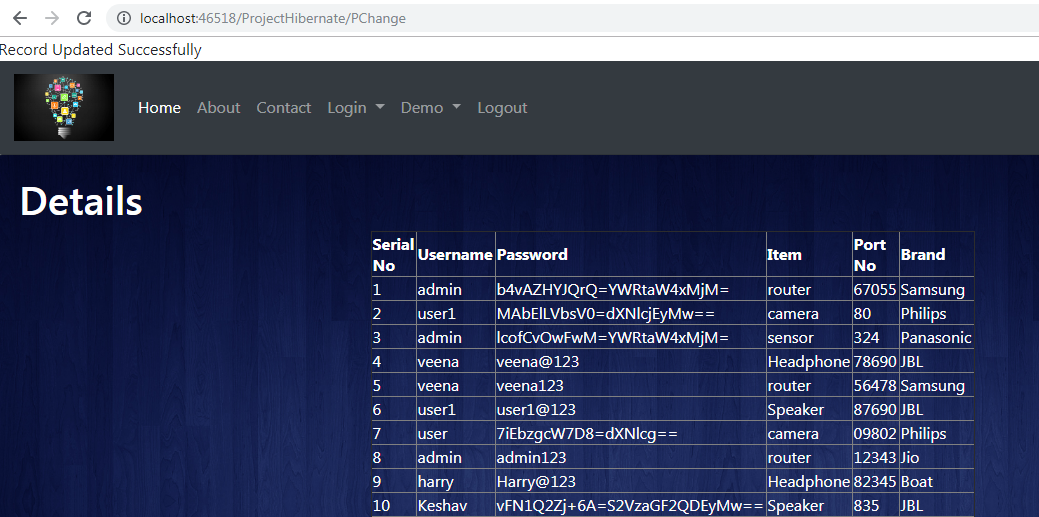


**Fig.5.8 Edit specific data**

When the admin clicks change button in the page, it leads to the above page. Here admin can edit the password of specific user by clicking the edit link present in the table.



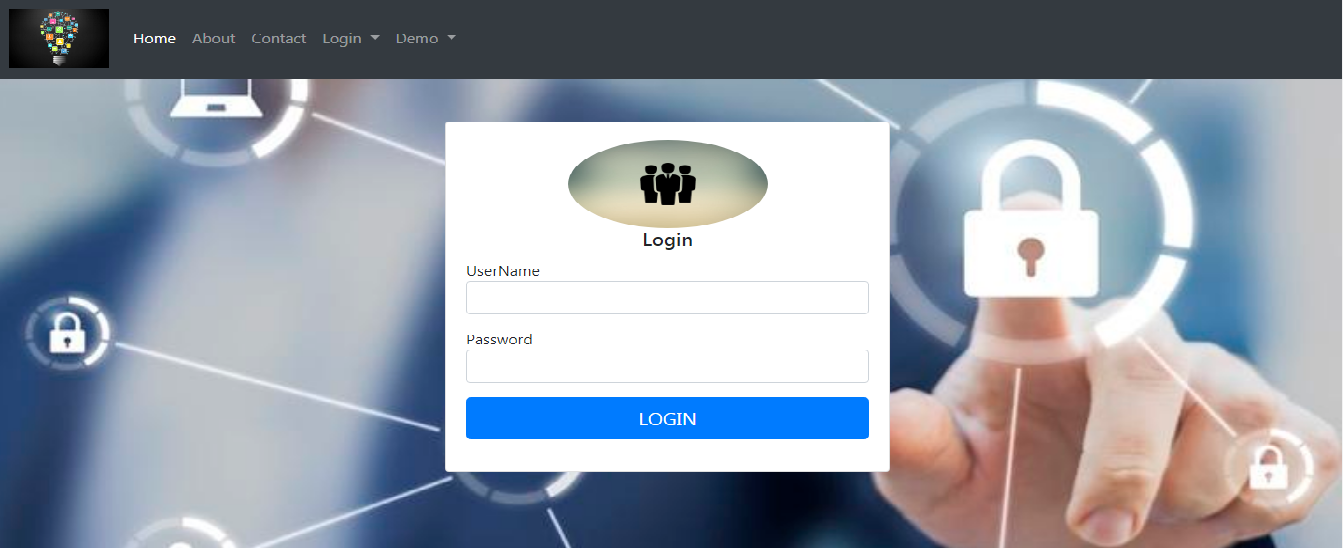
**Fig.5.9 Update data**



**Fig.5.10 Updated details**

You can see the updated password in the table. And the success message will be displayed. After doing all the operation, admin can click on the Logout link present in the navigation bar, which will logout and redirect to the Home page.

1. **Intelligence Team page**

****

**Fig.5.11 Intelligence team page**

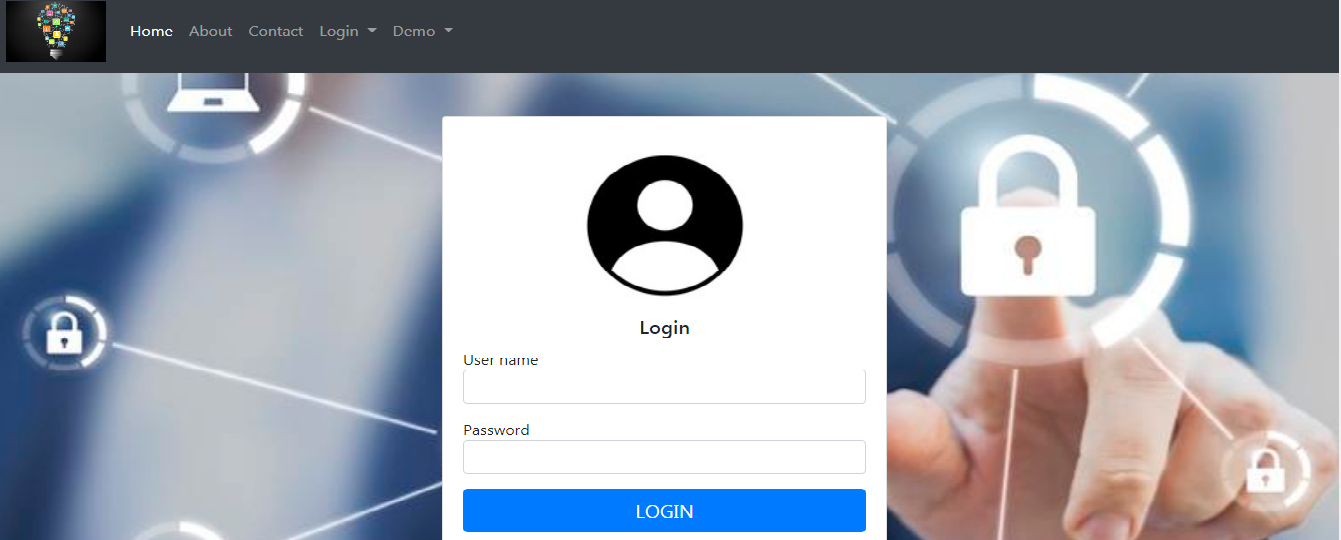
This is the login page for Intelligence team. This is similar to admin login page.



**Fig.5.12 Sensor sending data**

Once the intelligence team logs in, this page is visible. This page is the simulation of sensor sending data to the server. Here in the text area we can enter the data to be sent and in the dropdown list we can select the receiver’s name. When you click on send button, the message/data will be encrypted and sent to the server, where it will be stored as encrypted data.

1. **User page**

****

**Fig.5.13 User page**

This is the login page for User. This page is similar to both the previous admin and intelligence team login page.



**Fig.5.14 User receive page**

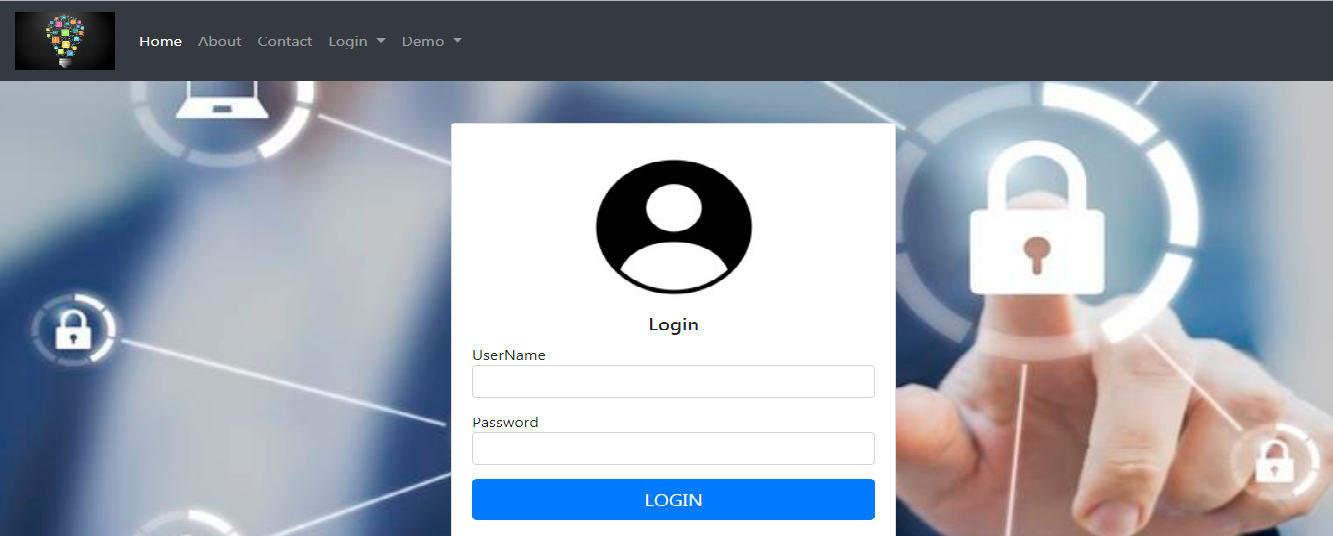
When the user logs in with correct username and password, he will get all the messages which is sent by the sensor and stored in the server. While sending the data to user, the server decrypts the encrypted data and sends it. There is a logout button inorder to logout from the page.

1. **Demo page**

****

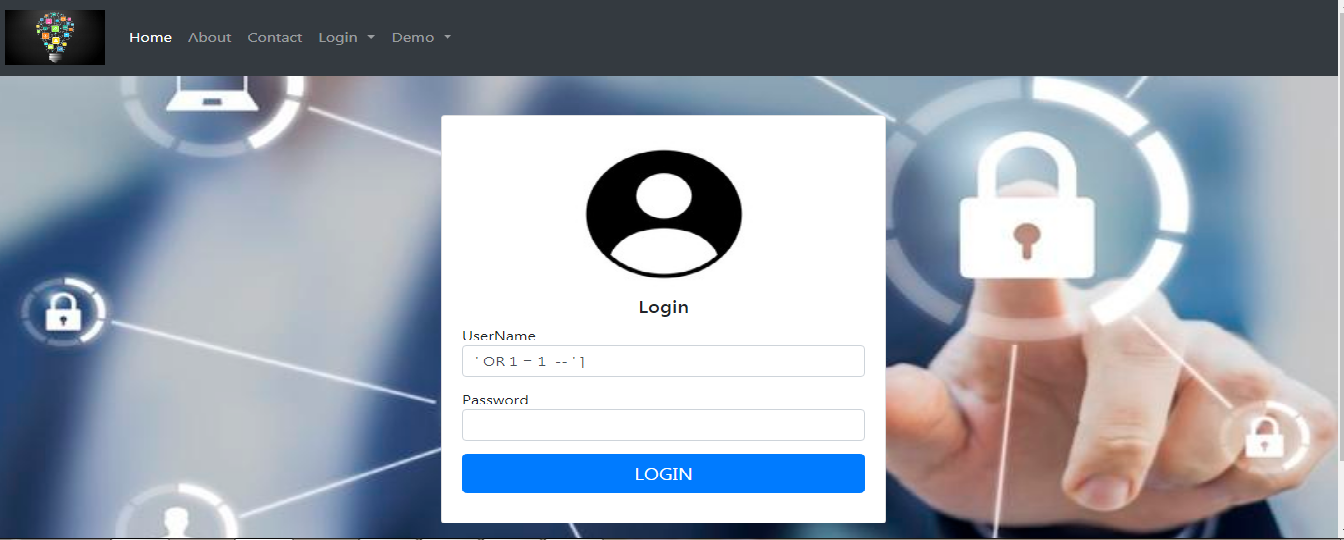
**Fig.5.15 Demo of sql injection**

In the home page you have one dropdown button called demo which will give you a demonstration of how SQL injection can be achieved and how we can prevent it.



**Fig.5.16 Vulnerable login page**

This is the vulnerable login page. Here, the attacker may use some SQL injection cheat codes inorder to get into the system and access the data.

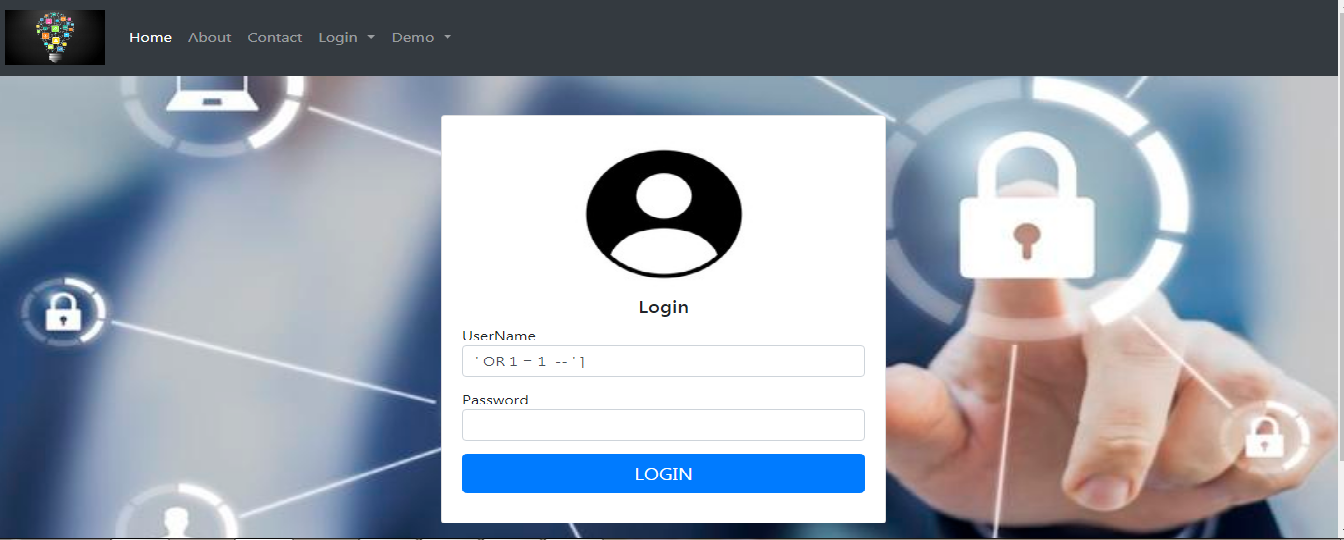


**Fig.5.17 SQL injection cheat code**

The above image shows the cheat code used by attacker. By using this cheat code and clicks on login button, the below page will be displayed.

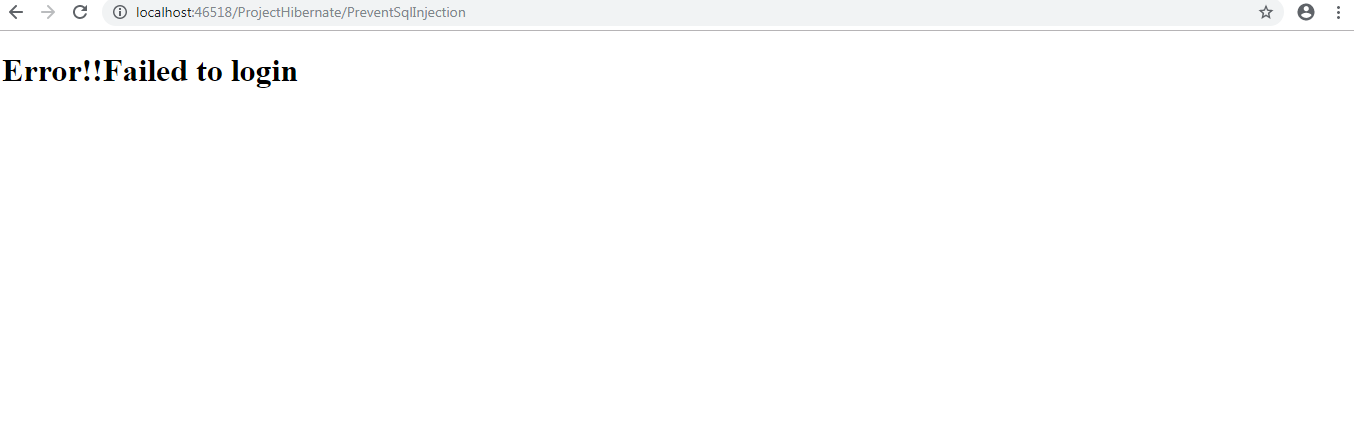


**Fig.5.18 Access permitted**



**Fig.5.19 Non vulnerable login page**

The above page is a non-vulnerable page. In this page if you try to attack it won’t allow you to make it.



**Fig.5.20 Access denied**

**6.1 TESTING**

**Introduction**

Testing is a process of executing a program with the explicit intention of finding error. It is a process used to identify correctness, completeness and quality of developed computer software. There are many approaches to software testing, but effective testing of complex product is essential a process of investigating. Testing helps in verifying and validating if he software working as it is intended to work. This involves using static and dynamic methodologies to test the application. There are two methods for test case design.

**6.1.1 White Box testing**

White box testing strategy deals with the internal logic and structure of the code. It is also called as glass, structural, open and clears box testing. The test that are written based on the white box testing strategy incorporate coverage of the code written, branches, statements and internal logic of the code etc. In order to implement white box testing the tester has to deal with the code and hence it is required possess knowledge of the coding and logic i.e. Internal working of the code.

**Advantages:**

* As the knowledge of the internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
* It helps in optimizing the code.
* It helps in removing the extra line of code, which introduce defect in the code.

**Disadvantages:**

* As the knowledge of code and internal structure is a prerequisite, a skilled tester is needed to carry out this type of testing, and these, in turn, increase the cost of the software.
* It is nearly impossible into every bit of code to find out the hidden errors, which may create problems, resulting in failure of application.

**6.1.2 Black Box testing**

Black box testing takes the internal perspective of the test object to derived test cases. These tests can be functional or non-functional though usually functional. The test designer selects valid and invalid inputs and determines the correct input. There is no knowledge of the test object’s internal structure. This method of test design is applicable to all levels of software testing: unit, internal, functional and system and acceptance.

**Advantages:**

* Black box test are reproducible.
* The environment in which the program is running is also tested.
* The invested effort can be used multiple times

**Disadvantages:**

* The results are often over estimated.
* Not all properties of the software can be tested.
* The reason for failure is not found

**6.1.3 Testing Methodologies**

* Unit Testing
* Integration Testing
* System Testing
* Validation Testing
* Output Testing
* User Acceptance Testing

**Unit Testing**

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage producers, are tested to determine if they are fit to use. Intuitively, one can view a unit as the smallest testable part of an application, In procedural programming a unit could be an entire module but is more commonly an individual function or procedure. In object oriented programming a unit is often an entire interface, such as class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process.

**Integration Testing**

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These design items, i.e. assemblages (or group of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test that all components with in assemblages interact correctively, for example across producers call of procedures activation, and this is done after testing individual modules, i.e. unit testing.

**System Testing**

A system testing of software or hardware is testing conducted on a complete, integrated system to evaluate system’s compliance with its specified requirements. System testing falls within the scope of black box testing, and such as, should require no knowledge of the inner design of the integrated software components that have successfully passed integration testing and also the software components itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between software units that are integrated together(called assemblages) or between any of the assemblages and the hardware, System is more limited type of testing, it seeks to detect defects both within the inter-assemblages and also within the system as whole.

**Validation Testing**

At the culmination of integration testing software is completely assemble as a package. Interfacing errors have been uncovered and corrected and fin; series of software test-validation testing begins. Validation testing can be defined in many ways but a simple definition is that validation succeeds when software functions in manner that is reasonably expected by the consumer.

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 specification that 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 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.

**Output Testing**

After performing the validation testing the next step is output testing of the proposed system since a system is useful if it does not produce the required output in the specific format required by them tests the output generator displayed on the system under consideration. Here the output is considered in the two ways- one is the onscreen and the other is printed format. The output formation the screen is found to be correct as the format was designed in the system design phase according to the user needs. As far as hardcopies are considered it goes in terms with the user requirement. Hence output testing does not result any correction in the system.

**User acceptance Testing**

User acceptance of the system is a key factor for 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.

**6.2 Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Input** | **Test Case Data** | **Expected Output** |
| 1 | User name and password | Correct user name and password | Login is successful |
| 2 | User name and password | Wrong user name and password | Redirect to the home page. |
| 3 | Username and password | Username and password are left blank | Redirect to the home page. |

**Test Case for Login:**

**Table 6.1 Test case for login**

**7.1 CONCLUSION**

This system is a security system for your IoT devices. It provides you security in three levels namely, device level, transmission level and storage level. The manufacturer of your IoT devices will have much interest in the profit obtained by selling the product. He never bothers about the security aspect and provides simple passwords. The Evaluation System for IoT will identify the weaker passwords and generate a random password which is highly secure. Similarly, the data can be accessed when it is transmitting over the channel. So, the Evaluation system for IoT will encrypt the data while it is transmitting by using Data Encryption Standards algorithm. And also, this system will secure your data from SQL injection.

**7.2 FUTURE WORKS**

The proposed system will generate the random password and stores it in database. Further research can be done on how to send the randomly generated password to the real user.

To increase the efficiency of the system, we can use other high level algorithms for encryption.

**REFERENCES**

[1] “Omner Barajas”, “How the Internet of Things (IoT) is changing the Cyber security Landscape”, “Security Intelligence”, 2014.

[2]Zlata Rodionova: TalkTalk given record fine over data breach that led to data theft of nearly 157,000 customers. In Proc. INDEPENDENT, Wednesday 5 October 2016.

[3]Alfred NG: Smart cities around the world were exposed to simple hacks. In cnet, August 10,2018

[4]Hacking IoT-A case study on DTH vulnerabilities https://www.secpod.com

# [5]IoT security vulnerability: A case study of a Web camera https://ieeexplore.ieee.org

[6]JakobJenkov:JavaCryptography http://tutorials.jenkov.com/javacryptography/index.html January 29,2018.