SQL INJECTION PREVENTION SYSTEM.

Submitted by:

Kavita Mourya [ ]

Avnish Yadav [ ]

Akash Singh [ ]

Month and Year:

May 2018

Under the guidance of

Ms. Radhika Desai

Submitted in partial fulfillment of the requirements for qualifying

B.Sc.(I.T.),

Semester-VI Examination.

University of Mumbai

Thakur College Of Science And Commerce,

Thakur Village, Kandivali (E). Mumbai-400101.

Thakur Educational Trust's (Regd.)
THAKUR COLLEGE OF SCIENCE & COMMERCE tcsc
UGC Recognized * Affiliated to University of Mumbai
(NAAC Accredited with Grade 'A' [CPGA-3.10] * ISO 9001:2008)

This is to certify that the project entitled "**SQL INJECTION PREVENTION SYTEM".** Undertaken at the "Thakur College Of Science And Commerce" by Ms. Kavita Mourya [ ], Mrs. Avnish Yadav [ ], Mrs. Akash Singh [ ] in partial fulfillment of B.Sc.IT degree (Semester-VI) Examination had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

It is further that they have completed all required phases of the project.

Signature of External Examiner **Signature of Internal Examiner**

Signature of Project Guide **Signature HOD/ In-charge/ Coordinator**

College Seal

PREFACE

As Part of our B.Sc.(IT.) curriculum and in order to gain practical knowledge we are required to make a Project on **SQL INJECTION PREVENTION SYSTEM**. The basic objective here behind doing this is to use the practical knowledge of programming to develop a working software system.

In this project we have created a website which helps the user to get all the computer hardware accessories easily. It help them to save their time in searching for the product here and there. It provides fully secured system to the user.

Doing this project helped to enhance the practical knowledge regarding the development of software system. It helped in understanding the development process involved in software. It also helped in building the team work management and leadership skills.

ACKNOWLEDGEMENT

Our successful completion of this project was more than just our desire to earn a valued degree. We would like to sincerely thank all those who have given us endless support and guidance for successful completion of this project.

We would like to be thankful to internal guide **Ms. Radhika Desai** and extend our hearty gratitude to **Dr. Santosh Kumar Singh** Coordinator, Head of Information Technology, for his kind, inspiring and illustrious guidance and ample encouragement. We would like to express our sincere thanks and deep sense of gratitude to our parents for their everlasting support.

We would like to thank all our teachers for making the facilities for the project as and when needed and providing for latest book information to develop the project. We would also like to thank all of them for helping in the implementation with the coding of the project.

We would also like to thank our staff of the college who let us to utilize college laboratory for learning as much time we needed. Last but not least we would also like to thank all our friends and colleagues for their support and invaluable help provided.

Page Index

Topic **Page No.**

Table Index

Figure Index

1. Introduction:

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc. Now a days, it is really hard to get some time to go out and get them by ourselves due to busy life style. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online.

It requires lots of time to travel to the particular shop to buy the goods. Since everyone is leading busy life now a days, time means a lot to everyone. Also there are expenses for travelling from house to shop. More over the shop from where we would like to buy some thing may not be open 24*7*365. Hence we have to adjust our time with the shopkeeper's time or vendor's time.

In order to overcome these, we have e-commerce solution, i.e. one place where we can get all required goods/products online. The proposed system helps in building a website to buy, sell products or goods(hardware) online using internet connection. Purchasing of goods online, user can choose different products based on categories , online payments , delivery services and hence covering the disadvantages of the system of self moving to shop and making the buying easier and helping the vendors to reach wider market. And also safe and secure system is maintained at every step of your purchase so that the credential will not be disclosed anywhere.

Hackers are prone to a newly attack known as SQL Injection attack wherein the hacker fires the vulnerable SQL query and tries to hack over database so prevention of this type of attack is essential. This can be achieved using a strong secured website.

1.1. Objective and Scope:

The cutting edge for business today is Electronic Commerce (E-commerce). Most people think E-commerce means online shopping. But Web shopping is only a small part of the E-commerce picture.

In addition, E-commerce includes business-to-business connections that make purchasing easier for big corporations. While there is no one correct definition of E-commerce, it is generally described as a method of buying and selling products and services electronically.

SQL Injection is the major factor affecting the database and the prime concern for upgrading technical business and organization and our project ensures for prevention of this action.

The primary goal of this website is to manage customer for online shopping. To sell computer accessories such as RAM, Hard drive, CD-ROM, Laptop, Monitor, CPU, Keyboard, Mouse, Printer, Scanner, Speaker, Graphic card, Motherboard, Laptop charger and Laptop battery etc. with valuable price.

This is very useful website for that customer who has their own residence for computer repair. Usually this type of customers place large amount of order of product because in their shop there is requirement of computer part. So computer accessories website becomes very popular rapidly. This is big advantage of this website.

Customer can get their demand product in cheap rate since they can compare to the price value.

1.2. Theoretical Backgrounds:

SQL Injection is a code injection technique that might destroy the database. SQL Injection is one of the most common hacking techniques. It is a placement of malicious code in SQL statements, via webpage input.

There are various types of SQL Injection Prevention Method built in the system according to the type of injection:

- **SQL Injection Based on 1=1 is Always True**

  Here the user enters the smart input rather than a wrong input. The SQL statement is always valid and will return ALL rows from the "Customers" table since OR 1=1 is always TRUE.

- **SQL Injection Based on ""="" is Always True**

  A hacker might get access to user names and passwords in a database by simply inserting "OR""=" into the username or password textbox.

- **SQL Injection based on Batched SQL Statements**

  Most database support batched SQL statement.

  A batch of SQL statements is a group of two or more SQL statements, separated by semicolons.

  For e.g.:

  A textbox (search product) takes the value of item to be searched along with the other formatted SQL query separated by semicolon(;)

  The language used to develop the project is **PYTHON**

- **DJANGO(A PYTHON PACKAGE):**

  Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source. Some features are:

Ridiculously fast:
Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure:
Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable:
Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

Various SQL Injection Prevention Parameters in Django:

- **Cross site request forgery (CSRF) protection:**

  CSRF attacks allow a malicious user to execute actions using the credentials of another user without that user's knowledge or consent.

  Django has built-in protection against most types of CSRF attacks, providing you have enabled and use it where appropriate. However, as with any mitigation technique, there are limitations. For example, it is possible to disable the CSRF module globally or for particular views. You should only do this if you know what you are doing. There are other limitations if your site has sub domains that are outside of your control.

  CSRF protection works by checking for a secret in each POST request. This ensures that a malicious user cannot simply "replay" a form POST to your website and have another logged in user unwittingly submit that form. The malicious user would have to know the secret, which is user specific (using a cookie).

  When deployed with HTTPS, **CsrfViewMiddleware** will check that the HTTP referer header is set to a URL on the same origin (including subdomain and port). Because HTTPS provides additional security, it is imperative to ensure connections use HTTPS where it is available by forwarding insecure connection requests and using HSTS for supported browsers.

- **extra()** and **RawSQL**:

  Django also gives developers power to write raw queries or execute custom sql. These capabilities should be used sparingly and you should always be careful to properly escape any parameters that the user can control. In addition, you should exercise caution when using **extra()** and **RawSQL**.

- **Session security:**

  Similar to the CSRF limitations requiring a site to be deployed such that untrusted users don't have access to any subdomains, **django.contrib.sessions** also has limitations.

- **Additional security:**

  While Django provides good security protection out of the box, it is still important to properly deploy your application and take advantage of the security protection of the Web server, operating system and other components.

  - Make sure that your Python code is outside of the Web server's root. This will ensure that your Python code is not accidentally served as plain text (or accidentally executed).
  - Take care with any user uploaded files.
  - Django does not throttle requests to authenticate users. To protect against brute-force attacks against the authentication system, you may consider deploying a Django plugin or Web server module to throttle these requests.
  - Keep your **SECRET_KEY** a secret.
  - It is a good idea to limit the accessibility of your caching system and database using a firewall.

1.3. Problem Definition:

- The previous system only made the product available but reviews from the customer and the feedback of the product was not added as a feature.
- The website was developed in such a way that once the product is bought then the customer did not had the right to replace the product because of unavailability of replacement option in the website.
- The website developed was poor in maintaining the customer detail. It could not provide the details of authorized user in an appropriate format which made the customer to face a huge problem.
- There was no security built in order to secure the user's data which lead the hacker to hack on the essential information of the user and use the information illegally.
- There was a huge time amount spend in going to each and every shop in order to get a single product in hand which is required. It took a long time to search each n every product individually
- No algorithm was used to get the encrypted form of data to be stored in the database which lead to the problem of database attack.
- Payment gateway developed was not secure and appropriate as demanded by the customer.
- It did not provide the user with the wish list where in user can wish for the product and get it into their cart so as to buy them in future.
- It was not much user-friendly because of which users faced a bit problem.

1.4. Proposed System:

- After using product customer can give their appropriate feedback to respective product. So that other customer can see feedback and take appropriate decision for choosing the product.
- Provide better replacement policy. If products are defective then that can be replace between 21 days from the date of delivery of product. The user have to contact the seller in order to get the replacement of the product.
- Customer can view their detail in a proper systematic format as specified in the website. All the details required by the customer can be easily viewed by them.
- Information about customer are store in the database with high level of security before delivery of product.
- Since customer can sit at one place and do the essential transaction it saves their time.
- Various algorithms are implemented in order to maintain high level of secure system and to store data in database in an encrypted format.
- Payment gateway is built with additional options and with well maintained security at that level.
- There is a option for the customer to get their wished product in their cart which can help them to buy in future as and when required.
- If products are available online. Then customer can compare different product in relation to their price, appearance, etc.

1.5. Hardware and Software Requirements:

1.5.1. Software Requirements:

The purpose of this SRS is to specify the requirements of the web based software application, which is an online shopping system. This Software Requirements Specification provides a complete description of all the functions and specifications of modules. Different software requirements are:

- Python3.6
- Django2.0
- SQlite
- Web Browser: Google chrome, Mozilla Firefox, etc.

The system shall display all the products that can be configured.

The system shall allow user to select the product to configure.

The system shall display all the available components of the product to configure

The system shall display detailed information of the selected products.

The system shall provide browsing options to see product details.

1.5.2. Hardware Requirements:

- System Architecture: Processor AMD64 and Intel EM64T
- Processors for Windows x64: 550 MHz minimum
- Physical memory (RAM): 2 GB (2048 MB)
- Disk space: 12 GB
- Video adapter: 256 colors
- Screen Resolution: 1024 X 768 minimum.

Primary memory of system should be of at least 2GB (2048 MB) so that programs execute at normal speed without any interrupt.

To display the content on monitor. Video adapter of monitor should support 256 color for representing the content in color combination.

1.6. Software Requirement Specification:

- **Introduction:**

In day to day life, we will need to buy lots of goods or products from a shop. It may be food items, electronic items, house hold items etc. Now a days, it is really hard to get some time to go out and get them by ourselves due to busy life style. In order to solve this, B2C E-Commerce websites have been started. Using these websites, we can buy goods or products online just by visiting the website and ordering the item online by making payments online with fully secured systematic way.

- **Purpose:**

The proposed system helps in building a website to buy, sell products or goods(hardware) online using internet connection. Purchasing of goods online, user can choose different products based on categories , online payments , delivery services and hence covering the disadvantages of the system of self moving to shop and making the buying easier and helping the vendors to reach wider market. And also safe and secure system is maintained at every step of purchase.

- **Scope:**

Customer can trust the system and carry on with digital shopping just by ensuring some of their details. He/ She can get verified an surf for what every they wish to get in the computer accessories. Prevention and protection is ensured at each stage therefore customer can very well shop with it which enhances the scope of the system.

- **Functional Requirement:**

System Interface:

SQL server is used as the database. Customer gets all the detail of the products and order with appropriate security maintained at each level.

User Interface:

Different user interface are provided by the system. For e.g.:Customer interface, Admin interface, etc.

Software Interface:

Software will depend on the security features provided by the Operating System and the programming language Python

Memory Constraints:

Memory Constraints will come into play when the size of SQL server database grows to a considerable size.

Operations:

The product shall have fail-safes to protect the database from being corrupted or accidentally altered during a system failure**.**

- **Non-functional Requirements:**

Reliability:

The system is much reliable that is it efficient stores the details of order, payment, customer without any data lost.

Availability:

The website will be always available to the customer with all security.

Security:

High level security was essential for the project therefore it is the important factor that was considered.

- **Operational Environment:**

Hardware Requirements:

- I3 Processor Based Computer.
- 1GB RAM
- 5 GB Hard Disk

Software Requirements:

- Windows 10.
- Python
- Django.

- **System Features:**

Input:

Takes input from Customer, Admin and stores in database.

Output:

Details of product and order to user/customer.

Storage:

The data is stored locally on the device and it uses SQLite server.

Search:

The user can search the desired product if wants.

2. Modules and their Description:

Various more module involved in the system are:

(For website development):

2.1. Search for product:

We offer the customer to search their desired product within the website. Also along with it website provides the facility to the customer to filter the product according to lowest or highest price, etc.

2.2. Purchase:

Purchasing of the product involves various payment option that customer can select as per as their convince. Customer needs to fill all the required detail before actual process of payment.

2.3. Feedback:

After purchasing and paying the amount of the product customer can actually give feedback to the required product they have bought. And also customer can rate the product they have purchased.

2.4. Cancel order:

Customer have been given the privilege of cancelling the order before delivery of their product.

2.5. Admin:

Admin module is developed for the purpose that the admin can access each and every control as and when required. For e.g.: Inserting product, etc.

(For SQL Injection Implementation and Prevention):

SQL Injection can be classified into three major categories-

In-band SQL: Type of attack where user uses same communicational channel to bth launch the attack and gather the result.

Inferential SQLi: Here attacker is able to reconstruct the databaseby sending payload, observing the web application's response and the resulting behavior of database server.

Out-of-Band SQLi: This is not very common, mostly because it depends on the features being enabled on the database server being used by the web application.

Few modules based on it which are implemented in the project depending on the types are:

2.6. Error-based SQL Injection :

It is the type where injection technique relies on error message thrown by database server to obtain the information about the database.

Prevention:

Errors are disabled on a live site, or can be logged to a file which has restricted access.

2.7. Union-based SQL Injection:

It is a type of injection technique that leverages the UNION SQL operator to combine two or more select query.

Prevention:

Checking the input provided and SQL select query should be build such that no vulnerable parameters can be passed using it.

2.8. Boolean-based(content-based) Blind SQLi:

It is an inferential SQL Injection technique that rely on sending an SQL query to the database which forces the application to return a different result depending on whether the query returns a **TRUE** or a **FALSE** result.

Prevention:

Using csrf token to prevent the data that is to be entered in database through forms**.**

2.9. Time-based Blind SQLi:

This type of blind SQL injection relies on the database pausing for a specified amount of time, then returning the results, indicating successful SQL query executing.

Prevention:

Throwing a time out error if the query demands for waiting a long time.

2.10. Out-of-Band SQLi:

Out-of-band SQL injection is not very common, mostly because it depends on features being enabled on the database server being used by the web application. Out-of-band SQL injection occurs when an attacker is unable to use the same channel to launch the attack and gather results.

Prevention:

We have used **CsrfMiddleWare** as the prevention technique through which the server interacts with the database securely.

2.11. Prevention of data in database by using encryption techniques to store data:

Using different algorithm(sha256) to encrypt the data in database so as to secure the precious data.

2.12. Securing protection by authenticating user through email by sending OTP:

OTP(One Time Password) can be the simple way to authenticate a customer or a user.

3. System Analysis And Design:

System design includes all activities, which help the transformation of requirement specification into implementation. Requirement specification specify all functional and non functional exceptations from the software.

Software design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.

Systems analysis is a method of studying a system by examining its component parts and their interactions. Structured data analysis (systems analysis) helps in analysing the flow of information within an organization with data-flow diagrams.

Systems design, the process of defining the architecture, components, and data of a system to satisfy specified requirements. Object-oriented analysis and design is an approach to analysis and design of an application, system, or business that emphasizes modularity and visual modeling. Service-oriented analysis and design is a method of Service-oriented modeling to design business systems. Structured analysis, methods in software engineering for converting specified requirements into software programs and hardware configurations. Structured systems analysis and design method is a systems approach to the analysis and design of information systems.

The major activity which were the part of analysis are:

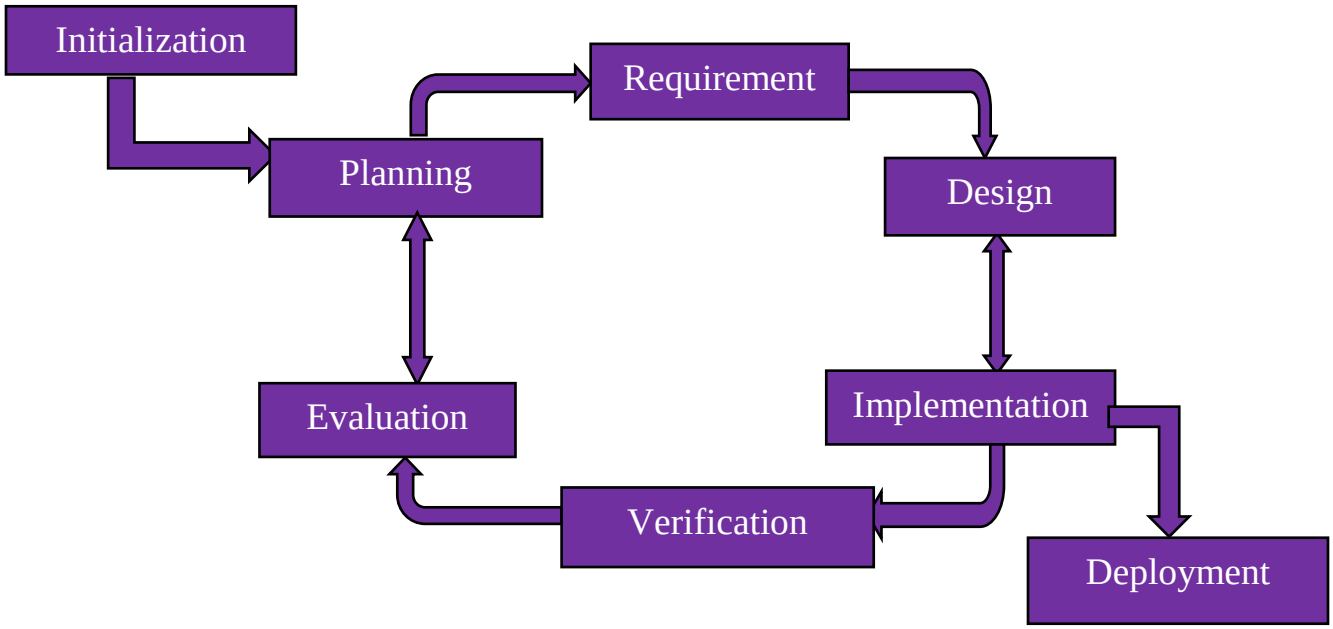
1. Collecting information for the system to be built.
2. Keeping track for all essential requirement to be covered.
3. Developing a prototype based on the requirement.
4. Managing the overall development.

3.1. Detailed Life-Cycle of the project:

Iterative model:

The iterative model is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.

Diagram:



After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last

Advantages of Iterative Model:

- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

Disadvantage of Iterative Model:

- Each phase of an iteration is rigid with no overlaps.
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle.

When to use Iterative Model:

- Requirements of the complete system are clearly defined and understood.
- When the project is big.
- Major requirements must be defined; however, some details can evolve with time.

3.2. Data Flow Diagram:

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. The DFD does not mention anything about how data flows through the system.

The development of DFD is done in several levels. Each process in the lower level diagram can be broken into a more detailed DFD in the next level. The top level diagram of DFD is called as context-diagram. This level is further detailed and elaborated at various stages or level.

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.[

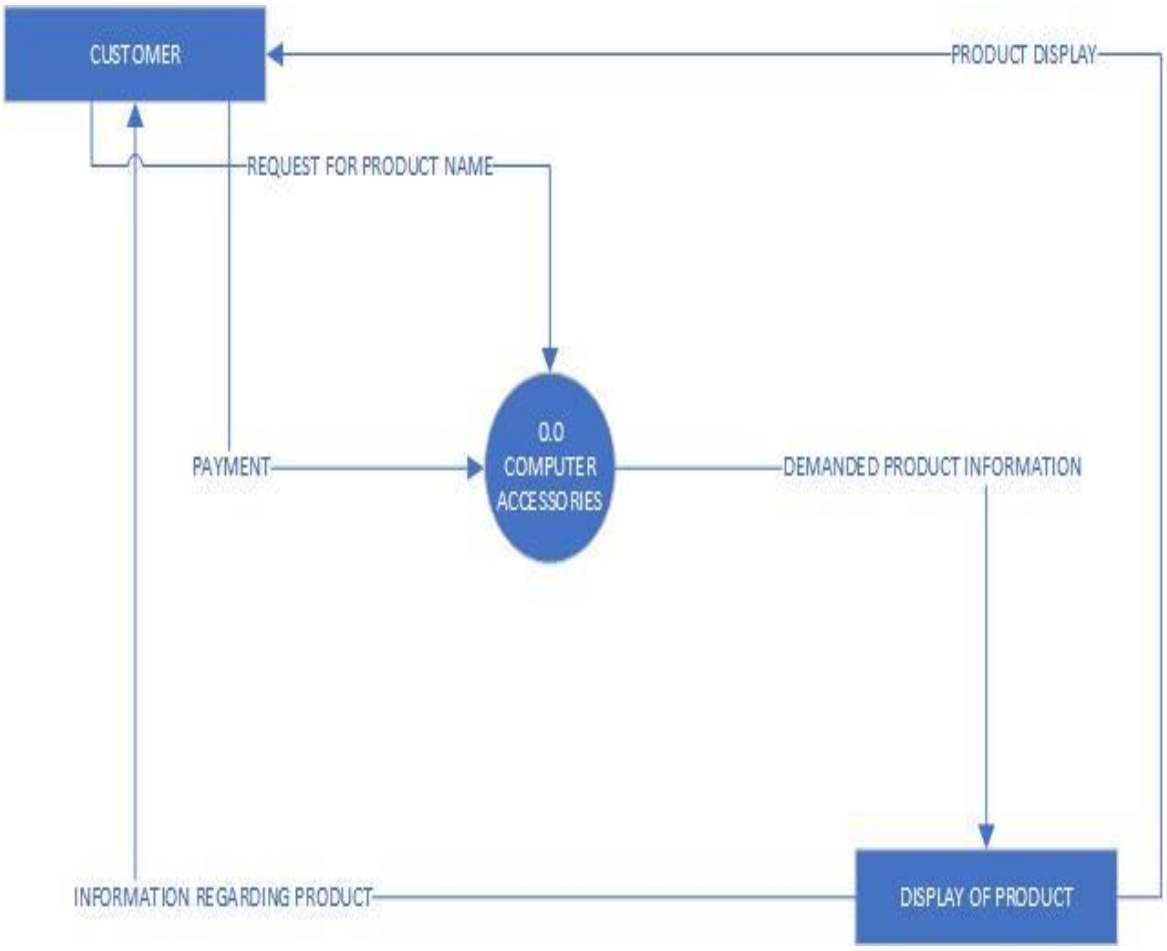
Data flow diagrams can be used in both the Analysis and Design phases of the SDLC.

How to create a data flow diagram:

1. Identify major inputs and outputs in your system.
2. Build a context diagram.
3. Expand the context diagram into a level 1 DFD.
4. Expand to a level 2+ DFD.
5. Confirm the accuracy of your final diagram.

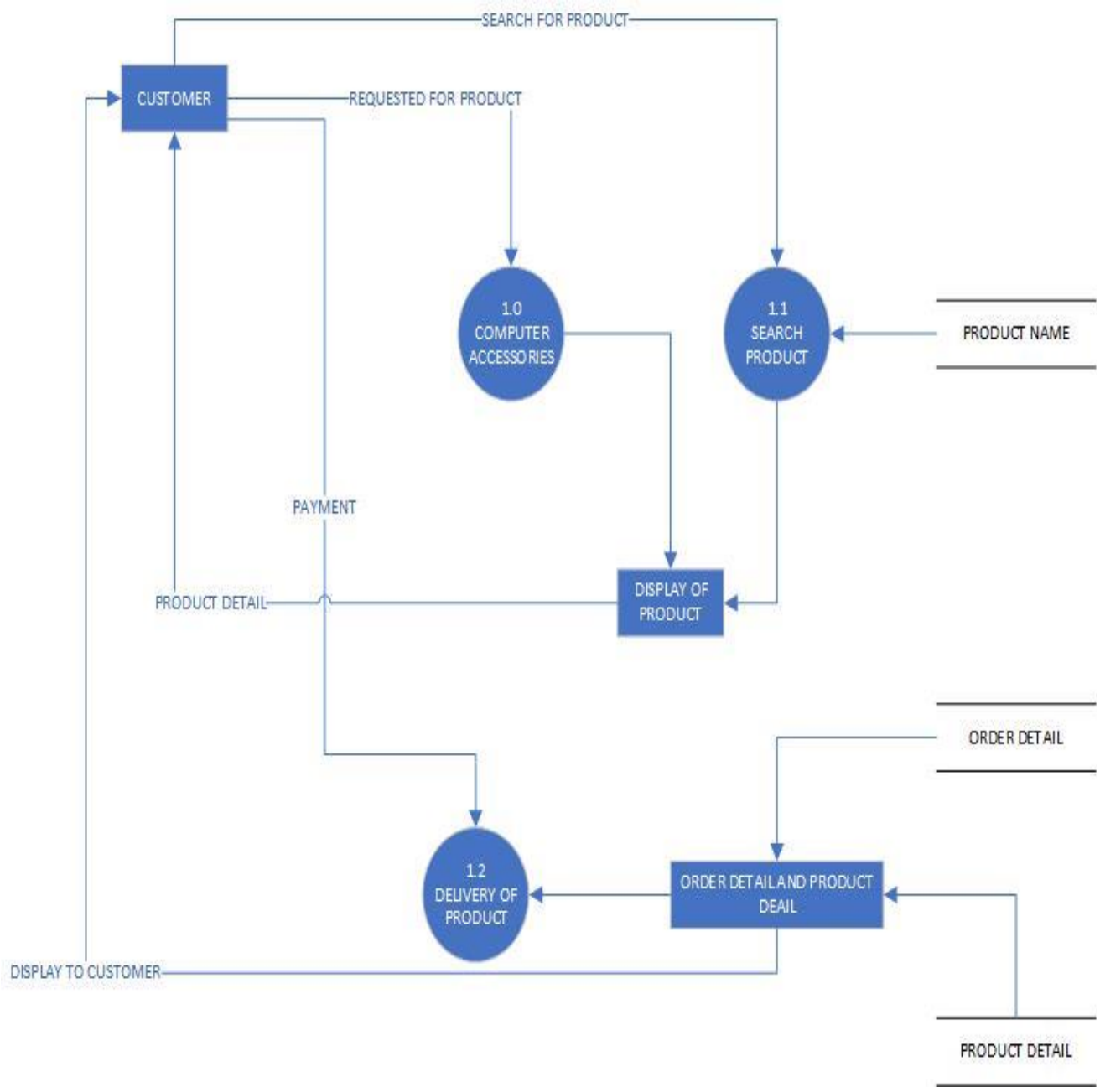
3.2.1. 0th level of DFD (Context Diagram):

Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.



3.2.2. 1st level of DFD:

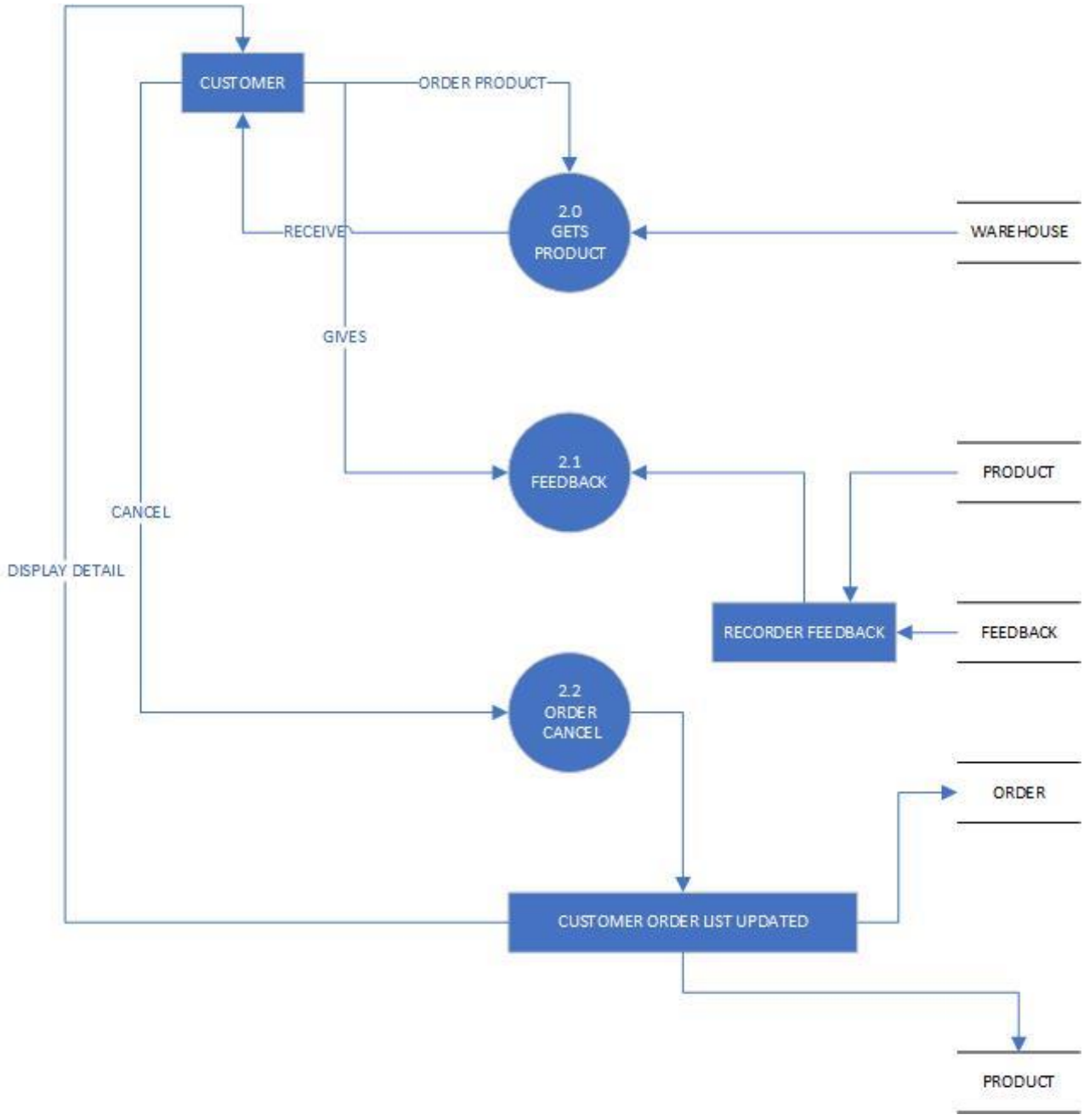
The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.



3.2.3. 2nd Level of DFD:

At this level, DFD shows how data flows inside the modules mentioned in Level 1.

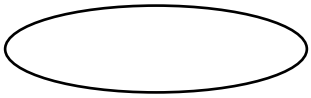
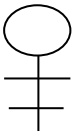
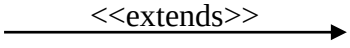
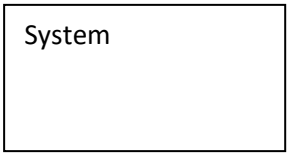
Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.



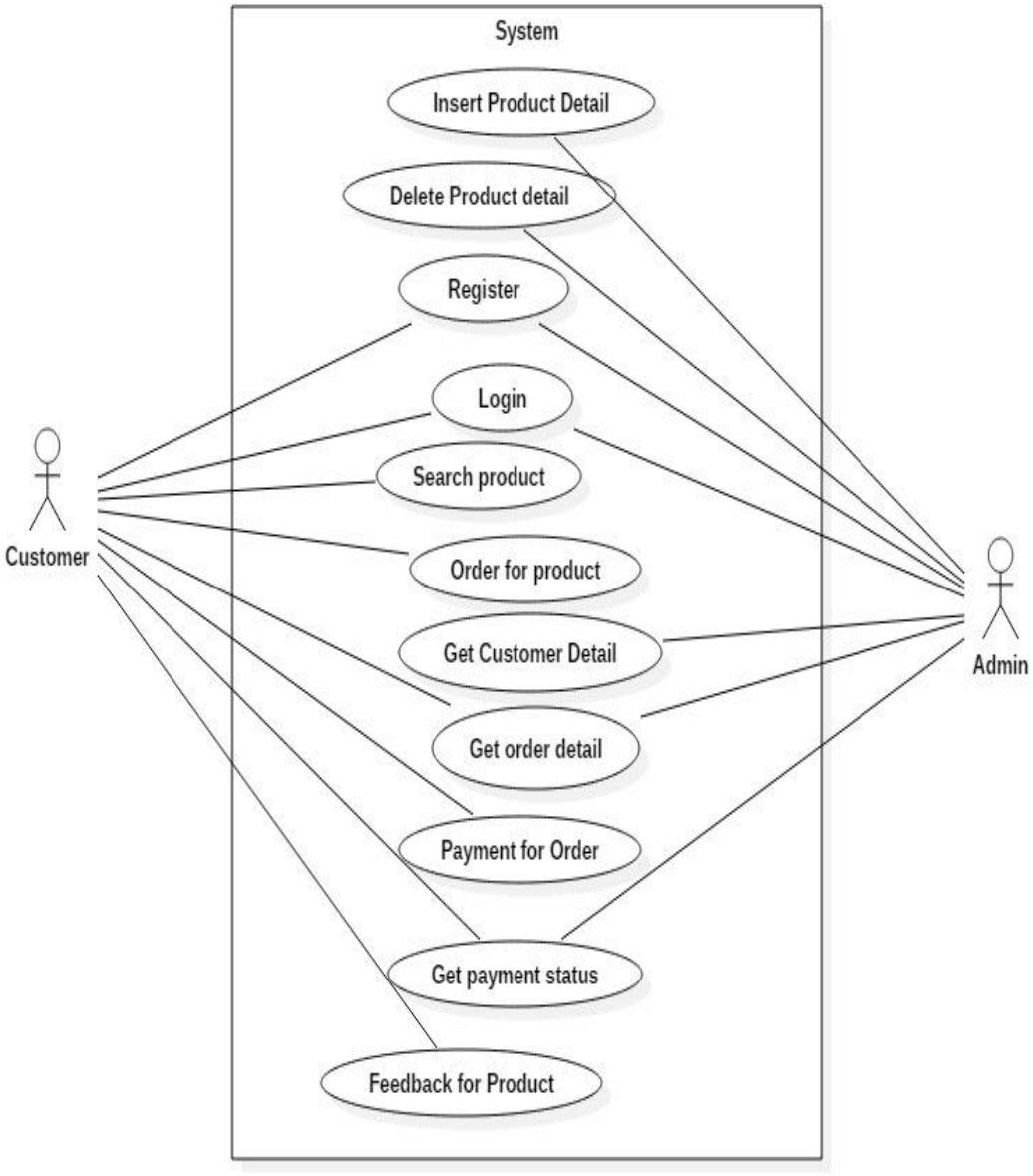
3.3. Use-Case Diagram:

A use-case diagram is the simplest representation of user's interaction with the system and also depicting the specification of use-case. A use-case diagram can portray different type of user's who interact with the system in different ways.

Various symbols used in a use-case are as follow:

Symbol	Description
 Use-Case	A use-case defines the squence of actionthat provides something of measureable values to an actor.
 Actor	An actor is a person, organization, or external system that plays a roles in one or more interaction with system.
<<include>>  <<extends>>  Relationship	Illustrate relationships between an actor and a use case. An "includes" indicates that one usecase is needed by the another usecase. An "extends" indicates alternative options under the certain use-case.
 System boundary	System boundary represents the scope of the system.

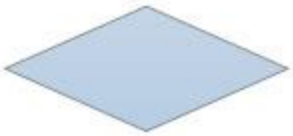
The proposed system use-case is as follow:



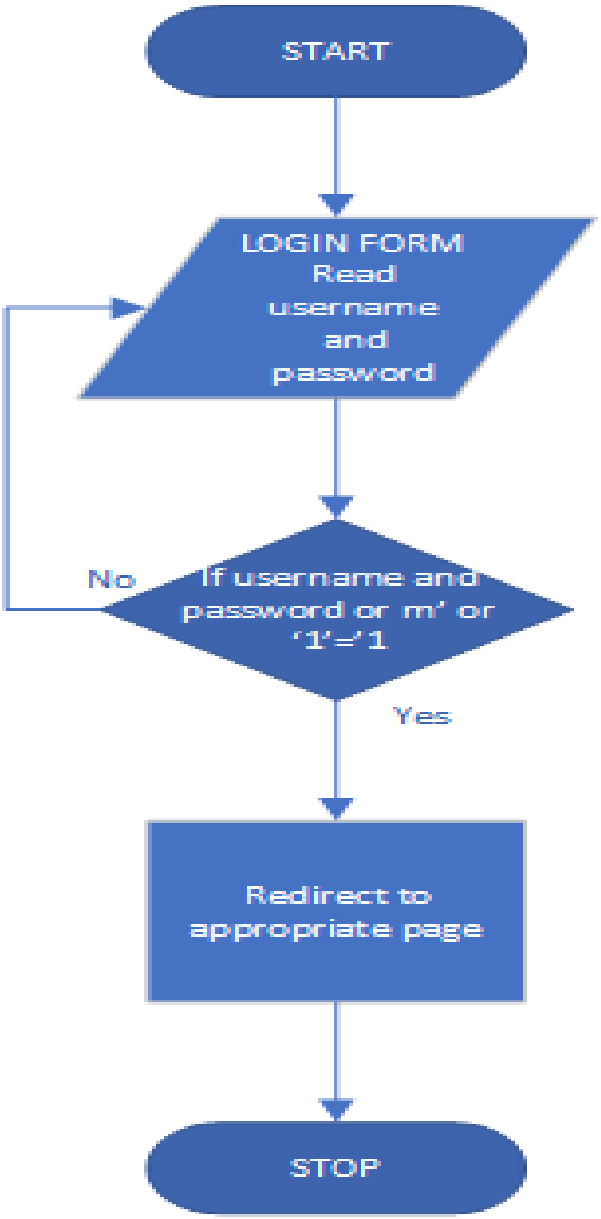
3.4. FlowChart For SQL Injection And Prevention:

Flow Chart is the graphically representation of a problem to be designed.

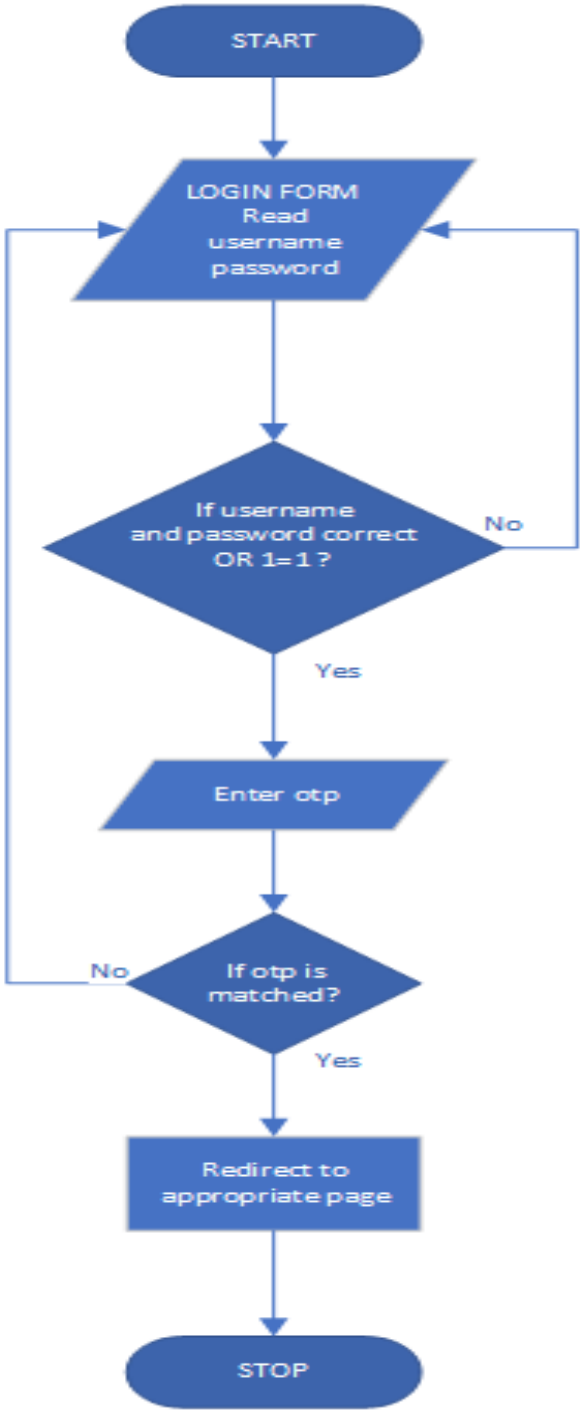
Various notation are used in developing a flowchart:

Symbol	Description
 Start	The terminator symbol marks the starting or ending point of the system. It usually contains the word "Start" or "End".
 Process	A box can represent a single step or entire sub-process within a larger process.
 Decision	A decision or branching point. Lines representing different decisions emerge from different points of the diamond.
 Input/Output	Represents material or information entering or leaving the system, such as customer order (input) or a product (output).
Control flow	It is used to indicate the overall flow of the flowchart.

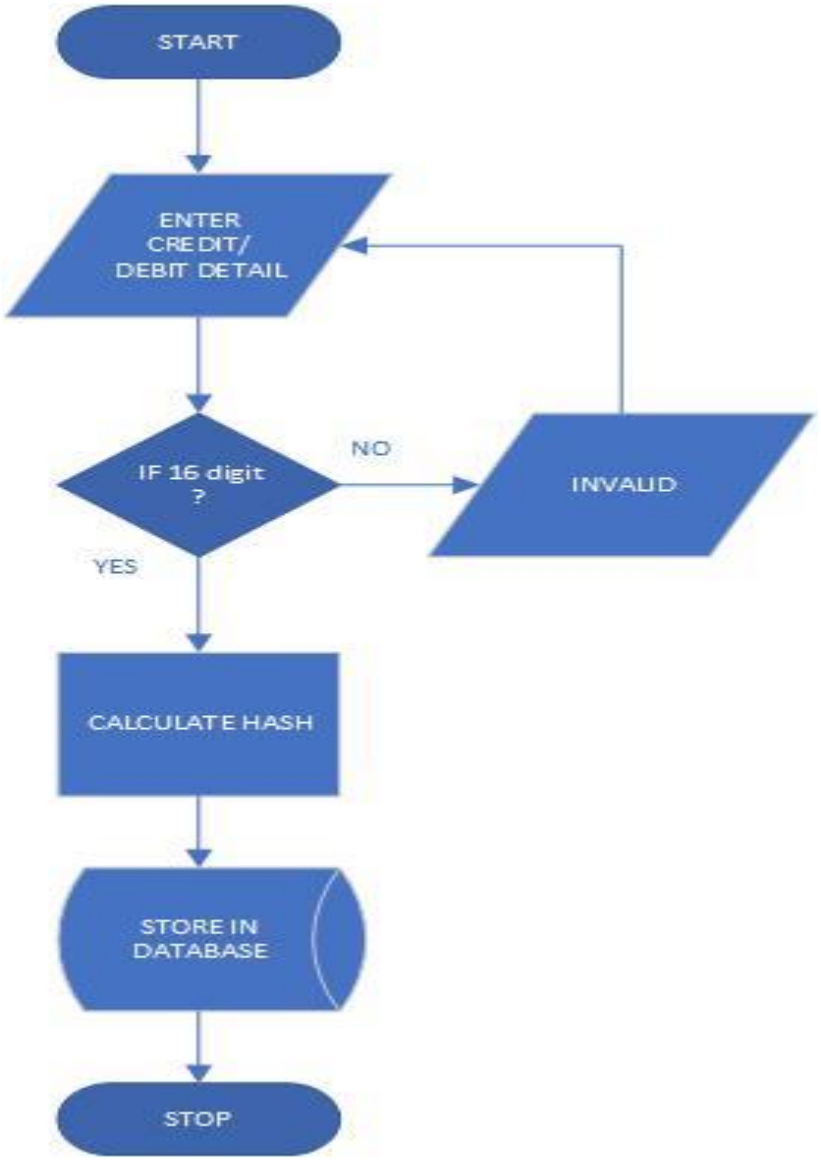
- **SQL Injection At Login Page:**



- **SQL Injection Prevention Technique At Login Page**:



- **SQL Injection Prevention at Payment Gateway:**



3.5. Entity-Relationship Diagram:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

The components and features of an ER diagram

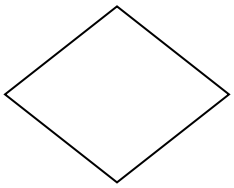
❖ Entity:

A definable thing such as a person, object, concept or event that can have data stored about it. Examples: a customer, student, car or product. Typically shown as a rectangle.

Entities are categorized as strong, weak or associative. A **strong entity** can be defined solely by its own attributes, while a **weak entity** cannot. An associative entity associates entities (or elements) within an entity set.

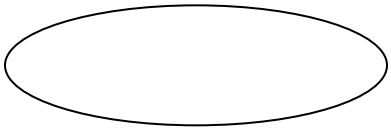
❖ Relationship:

How entities act upon each other or are associated with each other. For e.g.: The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.



❖ Attributes:

A property or characteristic of a relationship (versus of an entity) Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. **Simple:** Means the attribute value is atomic and can't be further divided, such as a phone number. **Composite:** Sub-attributes spring from an attribute. **Derived:** Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.



❖ Cardinality:

Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many.

3.6. Class Diagram:

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP).

The **purpose of class diagram** is to model the static view of an application. **Class diagrams** are the only **diagrams** which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row contains the attributes of the class, and the bottom section expresses the methods or operations that the class may use. Classes and subclasses are grouped together to show the static relationship between each object.

Basic components of a class diagram

The standard class diagram is composed of three sections:

- **Upper section:** Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.
- **Middle section:** Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.
- **Bottom section:** Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

3.7. Table Diagram:

Table Name	Fields	Datatype
Customer	CUSTOMER_ID	AUTOFIELD(FK)
	CUSTOMER_UNAME	CHARFIELD
	CUSTOMER_LNAME	CHARFIELD
	CUSTOMER_FNAME	CHARFIELD
	PHONE_NO	INTEGERFIELD
	EMAIL	EMAILFIELD
	DOB	DATETIMEFIELD
	PASSWORD	CHARFIELD

Table 1.1. Customer Table

Table Name	Fields	Datatype
Product	PID	AUTOFIELD(PK)
	QTY	INTEGERFIELD
	PRICE	DECIMALFIELD
	IMAGE	IMAGEFIELD
	IMAGE1	IMAGEFIELD
	IMAGE2	IMAGEFIELD
	NAME	CHARFIELD

Table1.2. Product Table:

Table Name	Fields	Datatype
Payment	PID	FOREIGNKEY
	CID	FOREIGNKEY
	OID	FOREIGNKEY
	CREDIT_NO	CHARFIELD
	DEBIT_NO	CHARFIELD
	CVV	INTEGERFIELD
	EXP_DATE	DATETIMEFIELD

Table 1.3. Payment Table

Table Name	Fields	Datatype
Order	OID	AUTOFIELD(PK)
	CUSTOMER_ID	FOREIGNKEY
	PRDT	FOREIGNKEY
	ORDER_DATE	DATETIMEFIELD
	AMOUNT	DECIMALFIELD
	QUANTITY	INTEGERFIELD

Table 1.4. Order Table:

Table Name	Fields	Datatype
Product_detail	PID	FOREIGNKEY
	FEATURE	CHARFIELD
	PRODUCT	CHARFIELD
	DESCRIPTION	CHARFIELD
	CAPABILITY	CHARFIELD

Table 1.5. Product_detail Table:

Table Name	Fields	Datatype
Product_Specification	PRDT_ID	FOREIGNKEY
	BRAND	CHARFIELD
	TYPE	CHARFIELD
	OTHER	CHARFIELD

Table 1.6. Product_Specification

Table Name	Fields	Datatype
Feedback	CID	FOREIGNKEY
	COMMENT	CHARFIELD
	RATING	INTEGERFIELD
	PRDT_ID	INTEGERFIELD

Table 1.7. Feedback

3.8. Architecture Diagram:

The **software architecture** of a program or computing system is the structure or structures of the system, which comprise **software** elements, the externally visible properties of those elements, and the relationships among them.

Software application architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application.

Like any other complex structure, software must be built on a solid foundation. Failing to consider key scenarios, failing to design for common problems, or failing to appreciate the long term consequences of key decisions can put your application at risk. Modern tools and platforms help to simplify the task of building applications, but they do not replace the need to design your application carefully, based on your specific scenarios and requirements. The risks exposed by poor architecture include software that is unstable, is unable to support existing or future business requirements, or is difficult to deploy or manage in a production environment.

While drawing an architectural diagram following things should be kept in mind are:

- Expose the structure of the system but hide the implementation details.
- Realize all of the use cases and scenarios.
- Handle both functional and quality requirements.
- Try to address the requirements of various stakeholders.

4. System Planning:

a. Gantt Chart:

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example: days, weeks, or months) and a vertical axis representing the tasks that make up the project. Horizontal bars of varying lengths represent the sequences, timing, and time span for each task. Gantt charts give a clear illustration of project status.

A **Gantt chart** is a graphical depiction of a project schedule. A **Gantt chart** is a type of bar **chart** that shows the start and finish dates of several elements of a project that include resources, milestones, tasks and dependencies.

Gantt-Chart for the proposed system is as follow:

Month / Activities	December	January	February	March
Communication				
Analysis				
Design				
Implementation				
Testing				

Table 2: Gantt-Chart

b. Process Involved:

Gantt charts give a clear illustration of project status, but one problem with them is that they don't indicate task dependencies – you cannot tell how one task falling behind schedule affects other tasks. The PERT chart another popular project management charting methods, is designed to do this. Automated Gantts charts store more information about tasks, such as the individuals assigned to specific tasks, and notes about the procedures. They also offer the benefits of being easy to change, which is helpful. Charts may be adjusted frequently to reflect the actual status of the projects task as, almost inevitably; they diverge from the original plan.

Planning Phase

We started planning phase in December 2017 and carried the procedure throughout. In this phase, we decided the flow of our project. We done Planning phase in Work Breakdown structure by dividing it into easily definable and understandable goals and tasks. A key element in the WBS is to plan for intended outcomes, rather than planning actions.

Requirement Analysis

We started this phase in December 2017 and ended in February 2018. We cleared all the requirements of the projects. Our requirement are as follows

1. Operating System : Windows 8 Professional

2. Environment : Python

3. Database : SQL Server

Design Phase

We started this phase in January 2018 and ended in March 2018. In this phase, we did design of the project by using different types of design tools and also we used Microsoft Word.

Coding Phase

We started this phase in January 2018 and ended in March 2018. In this phase, we did coding i.e. development of the project in Pycharm. Here, we designed our website in Python and also we connected to the database i.e. SQL server.

5. System Implementation:

Systems implementation is the **process** of: defining how the information **system** should be built (i.e., physical **system** design), ensuring that the information **system** is operational and used, ensuring that the information **system** meets quality standard (i.e., quality assurance).

The specific **implementation process** can vary from organization to organization, dependent largely on the details of the actual strategic plan, but some basic steps can assist in the **process** and ensure that **implementation** is successful and the strategic plan is effective.

Strategic implementation is critical to a company's success, addressing the who, where, when, and how of reaching the desired goals and objectives. It focuses on the entire organization. **Implementation** occurs after environmental scans, SWOT analyses, and identifying **strategic** issues and goals

The proposed system implemented on Windows is developed in various different phases. The implementation is done in such a way that each level of milestone is achieved successfully.

6. Cost And Benefit Analysis Model:

Cost-benefit analysis is one such concept that should be considered **a critical component of the software development process.** With an average failure rate of 20%, software projects are at a high risk for costing more money than they generate. Performing a thorough analysis of a project's costs and expected benefits or outcomes is the only way to identify whether or not a given project will be profitable and therefore viable for any company. A focus on quality products is a great asset to any business. However, without the ability to generate profits, the quality of your product offering has little value.

There are **three primary benefits** that smart businesses can enjoy from a cost-benefit analysis:

- **Loss Prevention:** When you can clearly see the costs that go into your software program and balance those with the sales profits, you will be able to prevent pouring more money into a product than you get out of it.
- **Increased profit:** Preventing a loss is important but it is in the generation of profits that your business can really succeed. A cost-benefit analysis can help to illustrate ways that your company can increase software sales, revenue and ultimately profits.
- **Improved decision making:** Every part of the software development process offers opportunities to streamline operations, reduce costs, or improve performance if the right information is made available. Having data readily accessible can help management and development teams make the right decisions at the right times.

In order to ensure that your cost-benefit analysis is truly useful, a cost-benefit analysis should factor in the following components:

- **Development costs**: This includes the time required of the development team, management and administrative staff. Any costs associated with tools, technology or other non-human resources should also be factored in.

- **Projected sales**: The identification of target sales goals should include expected sales and revenue volume at different price points to best identify the ultimate product price.
- **Sales costs**: When estimating sales revenue, any cost of selling your product should be included. This may include advertising, sales team commissions or salaries and more.
- **Maintenance costs:** Practically every software product that is developed will have ongoing corrective, adaptive or perfective maintenance tasks and associated costs. There are also planned updates or enhancements to the program that must be considered. Each of these costs should be factored into the original project from the outset.

Steps in the analysis:

Step 1:

The basic initial estimate for development effort is represented as:

$$E=A*(KSLOC)^{B} .$$

Where A and B are constants which are determined depending on type of the project. KSLOC is Thousand Single Lines Of Code.

Step 2:

Determine a set of 15 multiplying factors from different attributes attributes of the product which are:

Product attribute:

Required reliability, Product complexity, database size.

Computer attribute:

Execution Time Constraints, main storage constraints, Virtual machine volatility, Computer Turn-around Time

Personnel attribute:

Analyst Capability, Application Experience, Programmer Capabilities, Virtual Machine Experience, Programming Language Experience.

Project attribute:

Modern programming practice, Use of software tools, Required development schedule.

Each of the 15 attributes receives a rating on a six point scales that range from "very low" to "extra high". An appropriate efforts multiplier applies to the rating. The product of all effort multipliers results in an **Effort Adjustment Factor**(EAF). Typical value for EAF ranges from 0.9 to 1.4.

Step 3:

Final effort estimate is calculated by multiplying the initial estimate with EAF:

E=EAF*E (person-month).

Following is a basic COCOMO model of the system based on **ORGANIC MODE**

Table 3: Effort Adjustment Factor Table:

Product Attributes	Rating	Value
Required Software reliability	Very High	1.15
Complexity of the product	Very High	1.15
Hardware Attributes		
Run-Time performance constraints	Very High	1.10
Memory constraints	Nominal	1.00
Volatility of the virtual machine environment	Low	0.87
Computer turn-around Time	High	0.87

Personnel attributes		
Analyst Capability	Nominal	1.15
Application Experience	Nominal	1.00
Programmer Capability	Nominal	1.00
Virtual machine Experience	Low	0.87
Programming Language Experience	Very High	1.40
Project attributes		
Use of software tools	Nominal	1.00
Application Software Engineering Method	High	0.91
Required Development Schedule	Nominal	1.10

The Intermediate COCOMO model takes the form ……

$$\mathbf{E=EAF*A(KSLOC)^{B}.}$$

Where E is the Effort applied in person-month, KLOC is the estimated number of Thousands of Delivered Lines of codes for the project and EAF is the factor calculated above.

Here A=3.2 and B=1.05

EAF = 1.15*1.15*1.10*1.00*0.87*0.87*1.15*1.00*1.00*0.87*1.40*1.00*0.91*1.10

=1.543.

Estimation of Development Cost:

Total Time Required to Develop the project= 4-months

= 4*30

=120 days

Cost per day: 120

Number of developer:3

Total cost=4*30*120

Total cost=14,400/-

7. System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

System testing is a combination of both functional and non-functional testing. Functional testing, in simple terms, tests the functionality of whether an application is working to requirements or not. (smoke, sanity, retesting, regression, priority based testing, risk based testing etc.)

The steps involve in testing:

7.1. Unit Testing:

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

7.2. Integration Testing:

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

7.3. System Testing:

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

7.4. Validation Testing:

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

7.5. Performance Testing:

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

7.6. User Acceptance Testing:

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

7.7. Advantage of Project:

7.7.1. If products are available online. Then customer can compare different product in relation to their price, appearance, etc.

7.7.2. After using product customer can give their appropriate feedback to respective product. So that other customer can see feedback and take appropriate decision for choosing the product.

7.7.3. Provide better replacement policy. If products are defective then that can be replace between 21 days from the date of delivery of product.

7.7.4. Information about customer are store in the database with high level of security before delivery of product.

7.8. Disadvantage of Project:

May lead to unemployment of the worker and close down of the shops that sales the product.

7.9. Application:

- Buying product at secure system.
- Giving feedback so that the review helps other customer for that project.

7.10. Test-Cases:

A test-case is a document, which has a set of test data, preconditions, expected result, post condition, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test case act as a starting point of the execution, after applying the test values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

Modules	Test Name	Test Cases	Test Description	Observed	Remark
Customer	Validate access	Home page	Check if page is loads	Page loaded successfully	Tested
	Validate Contact	Contact	Check if page is loads	Page loaded successfully	Tested
	Validate Field	Login	Check if page is loads	Customer or admin logged in successfully	Tested

	Adding item validation	Wishlist	Check if page is loads	Product added to wishlist successfully	Tested
	Validate result	Logout	Check if page is loads	Logged out successfully	Tested
	Validate Search	Search button (for finding product)	Enter product name and click on search	Products are successfully searched	Tested
	Validate Button	Buy	Click	Products purchased successfully	Tested
	Validate button	Ok (for accept order)	Click	Order purchased	Tested
	Validate payment	Payment (buy different method)	Click	Payment successful	Tested
Admin	Validate Field	Login	Check if page is loads	Customer or admin logged in successfully	Tested
	Access Product List	Update Product List	Check if product list updated by add or delete of the product	Product List updated Successfully	Tested
	Access Order Detail	Retrieve order detail	Check if order detail retrieved successfully or not	Order detail retrieved successfully	Tested

Table 4: Test-Case

8. Feasibility Report:

Feasibility study deals with the cost analysis of the project development. Overall cost, time, technical cost, requirements, etc. Every cost-estimation is done.

It focuses on the major questions:

- What are the user's demonstrable needs and how does a candidate system meet them?
- What resources are available for given candidate system?
- What are the likely impacts of the candidate system on the organization?
- Whether it is worth to solve the problem?

The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

Steps in Feasibility analysis:

1. Prepare system flowcharts
2. Enumerate potential proposed system
3. Define and identify characteristics of proposed system
4. Determine and evaluate performance and cost effective of each composed system
5. Weight system performance and cost data
6. Select the best-proposed system
7. Prepare and report final application directive to management

8.1. Technical Feasibility:

It determines whether the technology under consideration is available and can be used effectively for the development of the application. The end user must be equipped with the pre-mentioned hardware and software requirements. It refers to the ability of the process to take advantage of the current stage of the technology.

The technical needs of the system may vary considerably. These may include:

- The facility to produce outputs in given time.
- Capacity of holding the required data.
- Acceptance of upgraded or developed system
- Guarantee of accuracy and reliability.
- Easiness of accessing the data and its security
- Existence of necessary technology.

Our project is technically feasible because all the technology needed for our project is readily available.

Operating System: Windows 10.

Language: Python(Django)

Database System: SQLite.

Documentation Tool: MS-Word 2010.

8.2. Economical Feasibility:

The economic feasibility of the system is mainly concerned with its financial aspects. It determines whether the project is economically feasible. Economic analysis is the most frequently used technique for evaluating the effectiveness of the proposed system. Most commonly known as Cost/Benefit analysis. They can be easily installed and implemented

In this issue, we should consider:

The cost to conduct a full system investigation.

The cost of h/w and s/w for the class of application being considered.

The development tools.

The cost of maintenance etc…

Our project is economically feasible because the cost of development is very minimal when compared to financial benefits of the application..

8.3. Operational Feasibility:

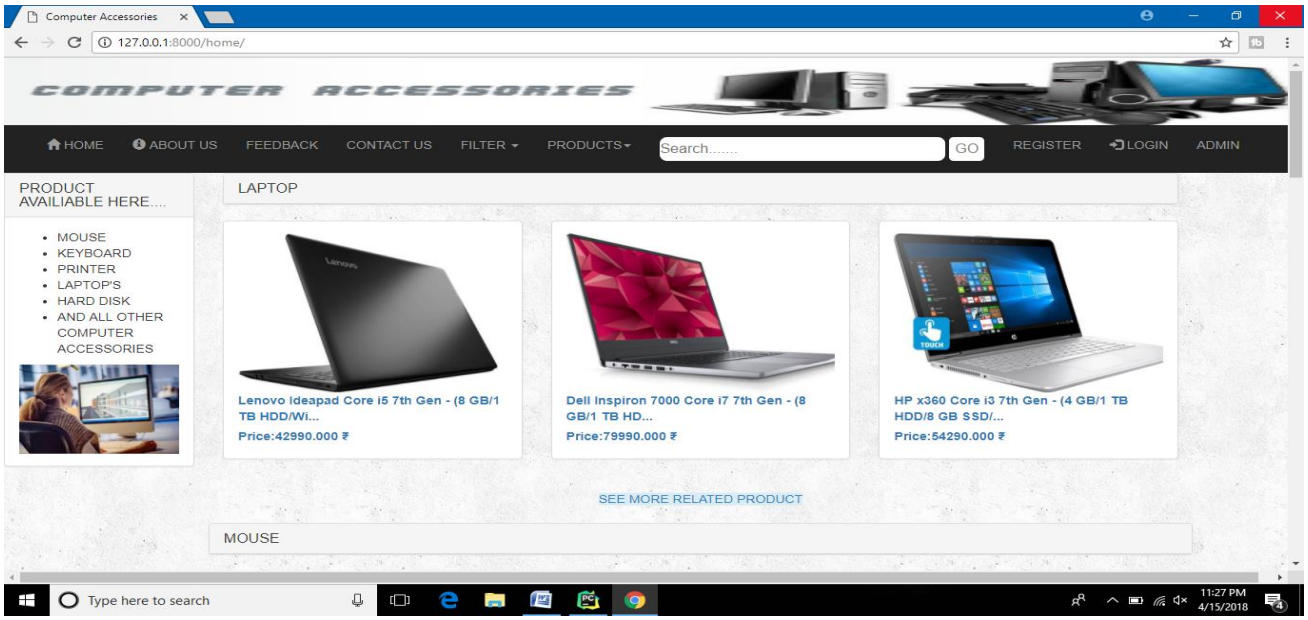
Once it is determined that the proposed system is both technically and economically feasible it has to be checked if that is operationally feasible. Operational feasibility is dependent upon determining human resources for the project. It refers to projecting whether the system will be operated and used once it is installed. There are 2 aspects to check the operational feasibility of the system. One is of the technical performance and the other is of acceptance. Technical performance deals with the fact that whether the system provides correct and timely data required for the End User.

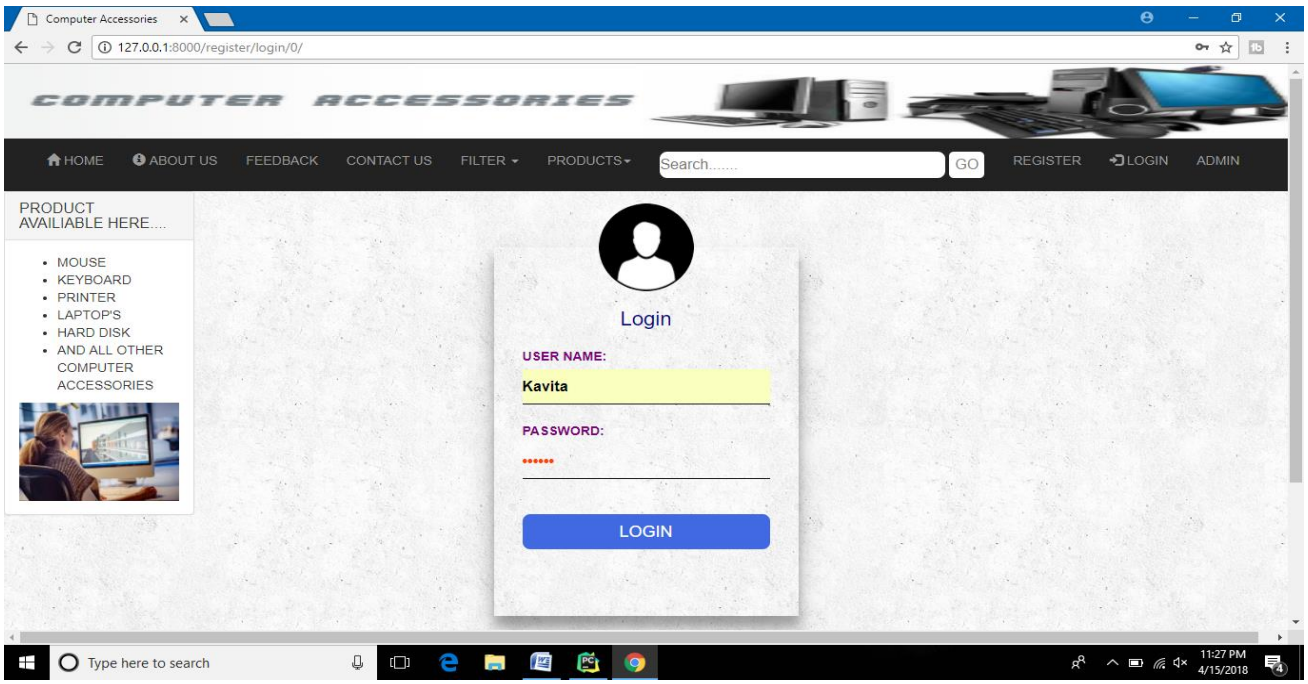
The operational feasibility mainly relates to the human, organizational and political aspects. The points to be considered are:

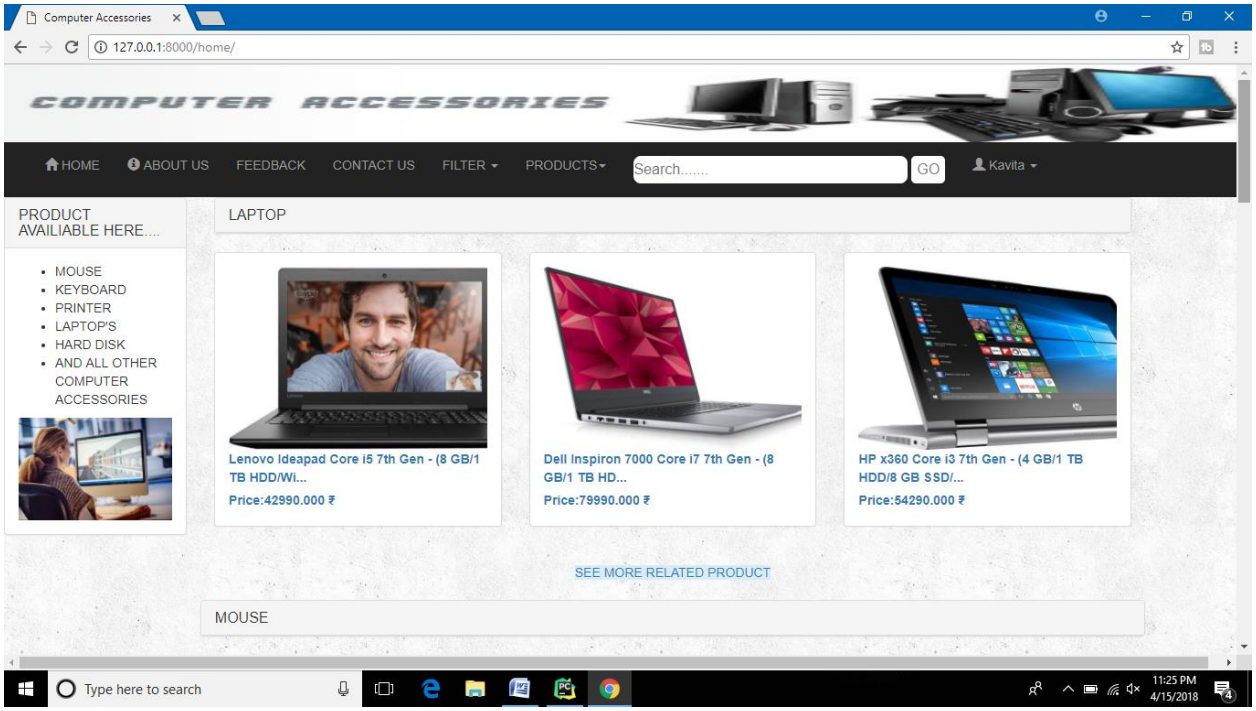
- What changes the application will bring?
- Whether the environment, in which system will be operated, will get disturbed?
- What new skills will be required?
- Whether the existing scorers will be able to operate the system?

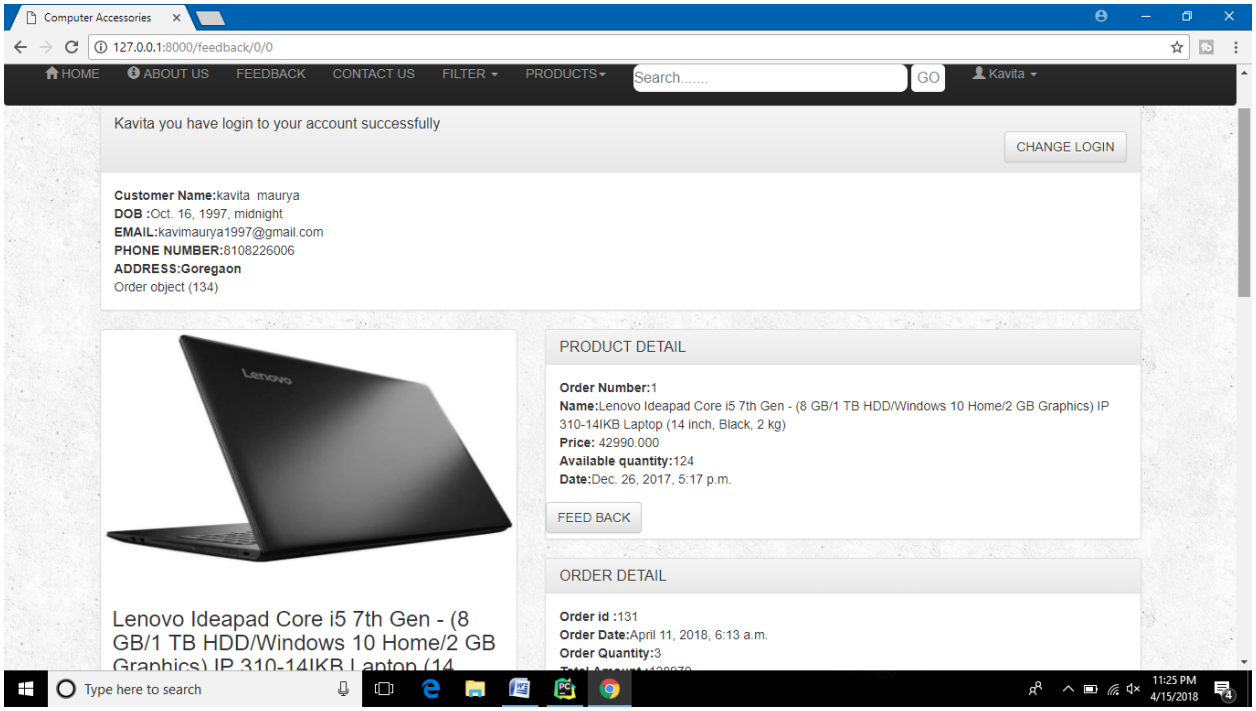
The proposed application will be much user-friendly which any user can easily operate after minimal amount of initial basic training.

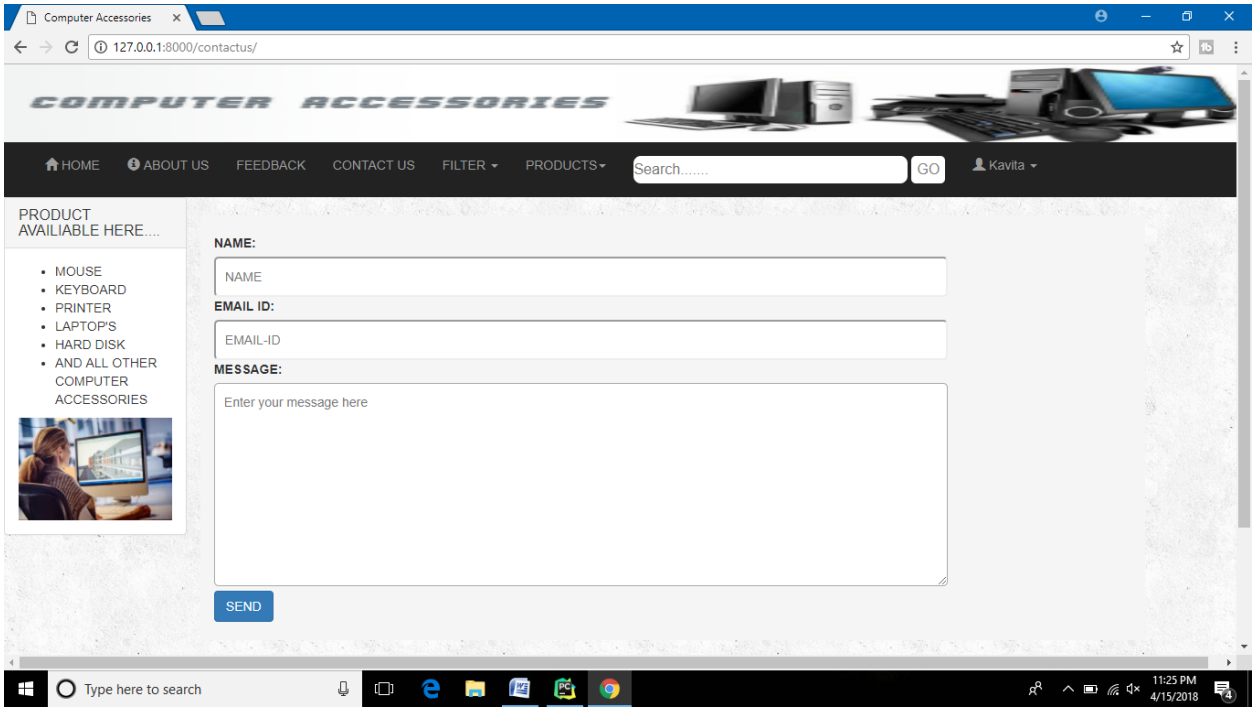
9. Screenshots:

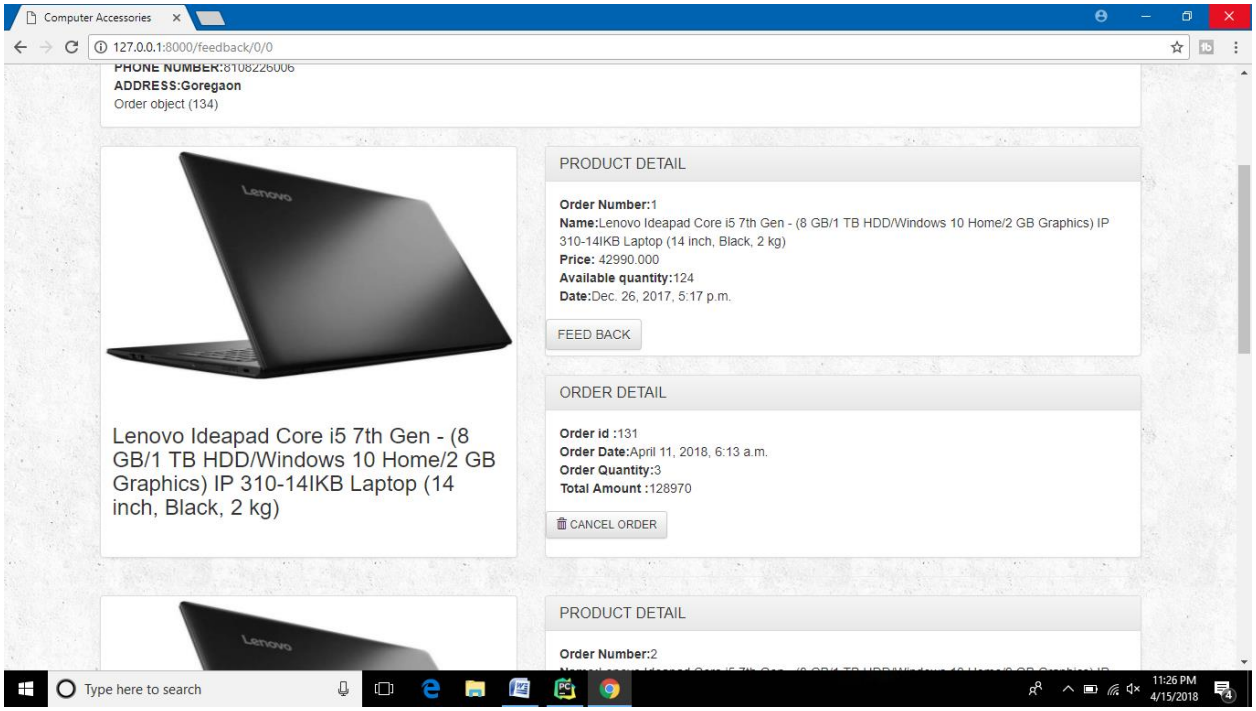





Computer Accessories
127.0.0.1:8000/home/
COMPUTER ACCESSORIES
HOME
ABOUT US
FEEDBACK
CONTACT US
FILTER
PRODUCTS
Search.....
GO
Kavita
PRODUCT AVAILIABLE HERE....
MOUSE
KEYBOARD
PRINTER
LAPTOP'S
HARD DISK
AND ALL OTHER COMPUTER ACCESSORIES
LAPTOP
Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Wi...
Price:42990.000 ₹
Dell Inspiron 7000 Core i7 7th Gen - (8 GB/1 TB HD...
Price:79990.000 ₹
HP x360 Core i3 7th Gen - (4 GB/1 TB HDD/8 GB SSD/...
Price:54290.000 ₹
SEE MORE RELATED PRODUCT
MOUSE
Type here to search
11:25 PM
4/15/2018


Computer Accessories
127.0.0.1:8000/feedback/0/0
HOME
ABOUT US
FEEDBACK
CONTACT US
FILTER
PRODUCTS
Search.....
GO
Kavita
Kavita you have login to your account successfully
CHANGE LOGIN
Customer Name:kavita maurya
DOB :Oct. 16, 1997, midnight
EMAIL:kavimaurya1997@gmail.com
PHONE NUMBER:8108226006
ADDRESS:Goregaon
Order object (134)
PRODUCT DETAIL
Order Number:1
Name:Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Windows 10 Home/2 GB Graphics) IP 310-14IKB Laptop (14 inch, Black, 2 kg)
Price: 42990.000
Available quantity:124
Date:Dec. 26, 2017, 5:17 p.m.
FEED BACK
ORDER DETAIL
Order id :131
Order Date:April 11, 2018, 6:13 a.m.
Order Quantity:3
Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Windows 10 Home/2 GB Graphics) IP 310-14IKB Laptop (14
Type here to search
11:25 PM
4/15/2018


Computer Accessories
127.0.0.1:8000/contactus/
COMPUTER ACCESSORIES
HOME
ABOUT US
FEEDBACK
CONTACT US
FILTER
PRODUCTS
Search.....
GO
Kavita
PRODUCT AVAILIABLE HERE....
MOUSE
KEYBOARD
PRINTER
LAPTOP'S
HARD DISK
AND ALL OTHER COMPUTER ACCESSORIES
NAME:
NAME
EMAIL ID:
EMAIL-ID
MESSAGE:
Enter your message here
SEND
Type here to search
11:25 PM
4/15/2018


Computer Accessories
127.0.0.1:8000/feedback/0/0
PHONE NUMBER:8108226006
ADDRESS:Goregaon
Order object (134)
Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Windows 10 Home/2 GB Graphics) IP 310-14IKB Laptop (14 inch, Black, 2 kg)
PRODUCT DETAIL
Order Number:1
Name:Lenovo Ideapad Core i5 7th Gen - (8 GB/1 TB HDD/Windows 10 Home/2 GB Graphics) IP 310-14IKB Laptop (14 inch, Black, 2 kg)
Price: 42990.000
Available quantity:124
Date:Dec. 26, 2017, 5:17 p.m.
FEED BACK
ORDER DETAIL
Order id :131
Order Date:April 11, 2018, 6:13 a.m.
Order Quantity:3
Total Amount :128970
CANCEL ORDER
PRODUCT DETAIL
Order Number:2
Type here to search
11:26 PM
4/15/2018

10. Conclusion And Scope for Future Work:

The proposed system is developed with the idea of implementing the overall secure system for the user accessing the website for particular transaction. The website is also developed with the motto of securing the data from vulnerable actions. Security of the system is the prime concern. User can purchase things they want without any issue of getting hacked by someone. Effort has been made in order to accomplish all user's need.

There is always a scope of improvement in any software, however efficient the system may be. So as for the scope of improvement it is essential that the software developed must be flexible to accommodate changes. Additional security can be very well added further with the proposed system now.

Some of the basic enhancements are:

- Using more packages that provides user interface can be built, can make the page more attractive.
- Searching can be improved by implementing effective algorithm.
- Since it is using iterative model any new requirement can be included.
- More secured system can be maintained using different techniques and technologies as and when essential
- More SOL Injection Prevention techniques can also be maintained and implemented.

Goals achieved:

The system is able to provide the interface to the owner so that they can replicate their desired data.

User-friendliness:

Efforts have been to make the website user-friendly so that every user can have access to the website without facing any problem. Effort have also been made to encrypt all the credentials of user to maintain security and trust.

11. References:

Website:

- www.stackoverflow.com
- www.djangoproject.com
- en.wikipedia.org
- www.w3school.com
- www.tutoralspoint.com
- www.quora.com
- www.owasp.org/index.php/SQL_Injection
- www.simpleisbetterthancomplex.com

Youtube:

- Cs geek tutorial
- ThenewBoston tutorial
- SQL Injection Implementation Types.

Research Papers:

- Wenlong Zhao, Junhu Zhu, Qingxian Wang, "Analysis and Prevention of SQL Injection", *Computer engineering and design*, vol. 27, no. 2, 2006.

Reference Book:

- Think Python.