**ABSTRACT**

The main objective of “LULU ONLINE JEWELLERY SHOPPING ” is to gain competitive advantage over the competitors who sale jewellery products. In this application, the shop keepers can easily handle their shop and business transactions by accessing single window. This proposed application helps customers to find and buy latest design of jewellery with different categories like Gold, Silver and Diamond based on their requirement. Customers can add the selected jewellery into the cart. The products that are added into the cart can be either purchased or deleted from the cart. Based on the purchase, customer can gain high discount and offers.

This application maintains the centralized database so that any changes done at any location reflect immediately. As it is online web application so more than one user can login into system and use the application simultaneously. This application can able to generate reports on sales, purchase and offers between a given time interval and it also provides a billing facility.

This web application will be developed by using HTML, JAVASCRIPT, BOOTSTRAP and CSS as front end. PYTHON FLASK framework will be used as middleware. MySQL will be used for database storage.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | **IV** |
|  | **LIST OF TABLES** | **VIII** |
|  | **LIST OF FIGURES** | **IX** |
|  | **ORGANIZATION PROFILE** | **X** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 OVERVIEW OF THE PROJECT | 1 |
| **2** | **SYSTEM ANALYSIS** | **2** |
|  | 2.1 EXISTING SYSTEM | 2 |
|  | 2.2 PROPOSED SYSTEM | 2 |
|  | 2.3 FEASIBILITY STUDY | 3 |
|  | 2.4 PROBLEM ANALYSIS | 4 |
|  | 2.4.1 PROBLEM DESCRIPTION | 5 |
|  | 2.5 DATA FLOW DIAGRAM | 5 |
|  | 2.6 SYSTEM SPECIFICATION | 12 |
|  | 2.6.1 HARDWARE REQUIREMENTS | 12 |
|  | 2.6.2 SOFTWARE REQUIREMENTS | 12 |
|  | 2.6.3 SOFTWARE DESCRIPTION | 12 |
| **3** | **SYSTEM DESIGN** | **17** |
|  | 3.1 INPUT DESIGN | 17 |
|  | 3.2 DATABASE DESIGN | 18 |
|  | 3.3 MODULE DESCRIPTION | 23 |
|  | 3.4 OUTPUT DESIGN | 24 |
| **4** | **SYSTEM TESTING** | **26** |
| **5** | **SYSTEM IMPLEMENTATION** | **29** |
| **6** | **CONCLUSION AND FUTURE ENHANCEMENTS** | **31** |

|  |  |  |
| --- | --- | --- |
|  | **APPENDICES** | **32** |
|  | APPENDIX 1 : SAMPLE CODE | 32 |
|  | **REFERENCES** | **51** |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF TABLES** |  |
| **TABLE NO.** | **TABLE NAME** | **PAGE NO.** |
| 3.1 | Admin | 18 |
| 3.2 | User | 19 |
| 3.3 | Product | 19 |
| 3.4 | Offer | 20 |
| 3.5 | Sales | 20 |
| 3.6 | Cart | 21 |
| 3.7 | Order | 21 |
| 3.8 | Placeorder | 22 |
| 3.9 | Invoice | 22 |
| 3.10 | Feedback | 23 |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** |
| 2.1 | Context Diagram | 6 |
| 2.2 | Level 1 DFD | 7 |
| 2.3 | Level 2 DFD for product details | 8 |
| 2.4 | Level 2 DFD for order placing and offer  details | 9 |
| 2.5 | Level 2 DFD for sales details | 10 |
| 2.6 | Level 2 DFD for Generating Bill | 11 |
| A2.1 | Home Page | 38 |
| A2.2 | Adding Products Form | 39 |
| A2.3 | Viewing Products | 40 |
| A2.4 | Offer Details | 41 |
| A2.5 | Registration | 42 |
| A2.6 | Viewing Products | 43 |
| A2.7 | Cart | 44 |
| A2.8 | Order Summary | 45 |
| A2.9 | Billing | 46 |
| A2.10 | Sales Report | 47 |
| A2.11 | Stock Report | 48 |
| A2.12 | Offer Report | 49 |
| A2.13 | Feedback | 50 |

## CHAPTER 1

**INTRODUCTION**

* 1. **OVERVIEW OF THE PROJECT**

The main goal of lulu online jewellery shopping System is to create shopping cart, which allows customers to shop and purchase the Jewellery products online with certain offers based on the number of times they have purchased. Moreover, the project is also designed in such a way it lets the admin to manage the jewellery products information.

In today’s busy world, people don’t have time for their personal needs. And the technology fast that anyone can do by sitting in a room. If someone buys new things, he can buy in online with the help of Internet.

The application is implemented in Python Flask and consists of two main components:

* + - Admin side and
    - Customer side

*Admin side* consists of the features such as Creating Username & Password, Input Items and Offers, Modify Items and Offers, Delete items and Offers, Query Sale Data, Query Database’s data, and Logout.

*Customer side* consists of the features such as Creating an Account, Sign-in, Select Products, Search Products, View Products, Buy Products, Continue Shopping, View Cart, Checkout, Confirm, Bill Information and logout.

## CHAPTER 2

## SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

In the existing system, the customer should purchase the product manually after placing the order and there are no promotions and offers given to the customers. The existing system does not generate bill with discount price and offers. The data is also not maintained efficiently which leads to double entries.

## Problem with Existing system

* + - No good customer service
    - There is a possibility for double entries
    - Generating the reports in the desired format is a tedious process
    - Reporting to the higher officials is not done through the proper channel

## PROPOSED SYSTEM

This web application provides services to the user for purchasing the jewellery products. The promotion will be given to the customers by recognizing them using some strategies such as sales promotion for all customers like free silver coin for every purchase, personalised discount (for loyal customers). It helps in tracking the customers and allows to provide customer satisfaction with the services. It generates report by accessing information stored in database.

## Advantages of the proposed System

* + - Improvement in customer service levels
    - Better sales performance
    - Enhanced quality in shopping experience

## FEASIBILITY STUDY

One of the many stages in the system development life cycle is determining the feasibility of a project. A measure of how beneficial or practical the development of a software system will be to an organization. This analysis recurs, and how that technology can be of benefit. These studies can include policy recommendations for imaging systems, document management, and workflow applications for an interactive process of collecting and analysing through the life cycle. Feasibility studies examine the organization data with the client and searching for cost-effective, viable technical and non-technical solutions. Three key considerations involved in feasibility analysis are

* + - Technical Feasibility
    - Economical Feasibility
    - Operational Feasibility

## Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.

The technical requirements of the application are simple and basic. The Python Flask is used for the development of the application and the framework is largely used by many, thus there will be enough support for future enhancements. The framework is stable and the supports from the developers are constantly updated. The devices which have a web browser and internet connectivity are enough for the application as it is a web service. This project can also be used via smart phones as it is a normal web service. To use the application in the smart phone, only a normal web browser is needed.

## Economical Feasibility

The economic analysis of one project counts for the cost effectiveness of the project. Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. This is also called as cost benefit analysis.

The cost of application development is very less and the cost of implementation is also less. It can be developed with the system with minimum requirements and can also be operated with the system with some basic requirements that are available in most of the existing systems.

## Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed.

The system is operationally feasible because a person with basic computer knowledge can access this application. As the system is developed in a user friendly manner, it is easily adapted by the everyone.

## PROBLEM ANALYSIS

The problem analysis is the process of understanding the real-world problems and user needs and proposed abstract solutions to those problems. It also provides a necessary stepping stone leading to the development of the software requirement specification.

Basically, it looks at the whole problem rather than trying to describe the system functions, and it uses a technique called “problem decomposition” to tackle the problems. As a result, it produces a much more understandable set of requirements.

## 2.4.1 Problem Description

The problem involved in lulu online jewellery shopping system is that there are no promotions and offers provided to the customers. The generation of reports such as purchase, sales and offer to administrator is a difficult process. There is no good customer services provided. This web application will provide a good customer service and promotions to the customer.

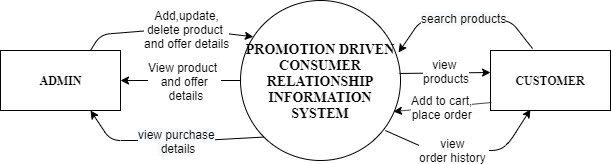
## DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.

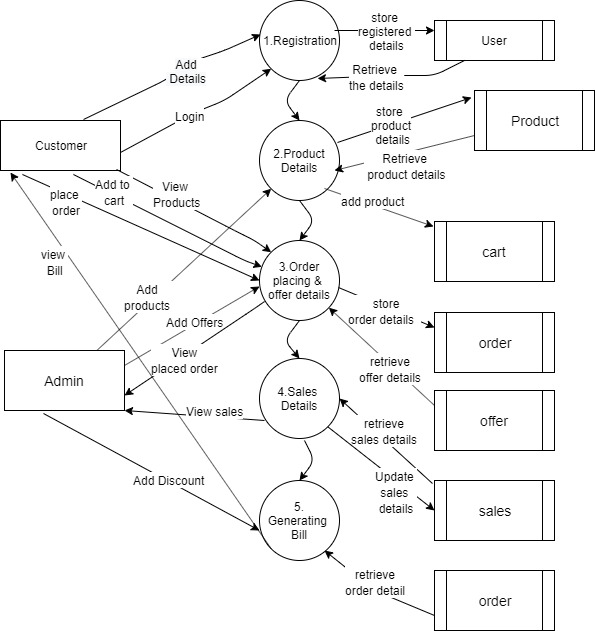
## DFD SYMBOLS

In the DFD, there are four symbols:

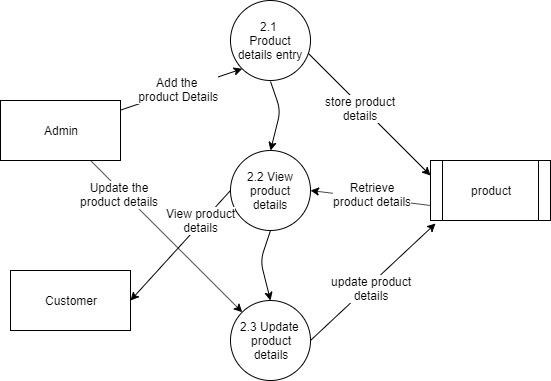
* + - A square defines a source(originator) or destination of system data
    - An arrow identifies data flow. It is the pipeline through which the information flows
    - A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
    - An open rectangle is a data store, data at rest or a temporary repository of data



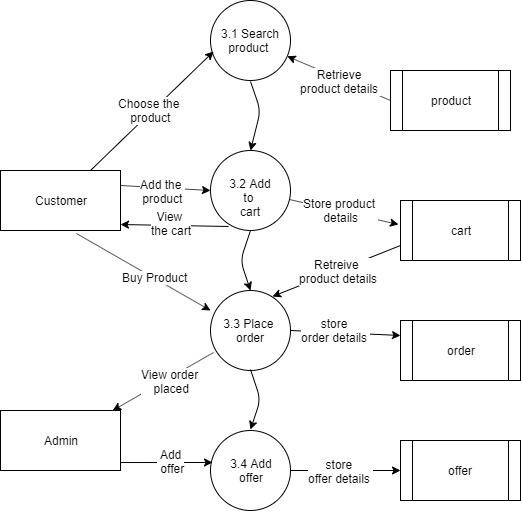
## FIGURE 2.1 CONTEXT DIAGRAM



**FIGURE 2.2 LEVEL 1 DFD**

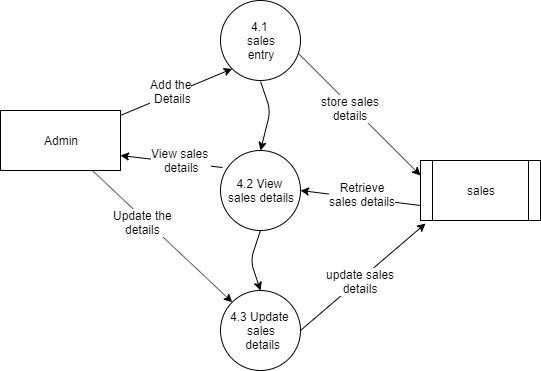


**FIGURE 2.3 LEVEL 2 DFD FOR PRODUCT DETAILS**

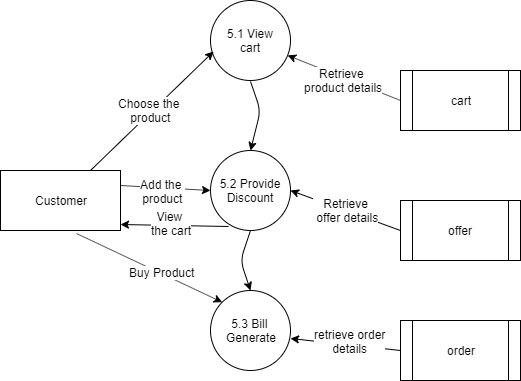


**FIGURE 2.4 LEVEL 2 DFD FOR PLACING ORDER**

**AND OFFER DETAILS**



**FIGURE 2.5 LEVEL 2 DFD FOR SALES DETAILS**



**FIGURE 2.6 LEVEL 2 DFD FOR GENERATING BILL**

* 1. **SYSTEM SPECIFICATION**
     1. **Hardware Requirements**

This section gives the details and specification of the hardware on which the system is expected to work.

Processor : Intel Dual Core or higher

RAM : 2 GB or above

Hard Disk : 300 GB or above

## Software Requirements

This section gives the details of the software that are used for the development.

Operating System : Windows XP or Higher Environment : Python Flask

Scripting language : HTML, CSS, Bootstrap Database : MySQL

Server : XAMPP

## Software Description

## HTML

HTML stands for HYPER TEXT MARKUP LANGUAGE, which is most widely used language on web to develop web pages. HTML refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a web page is called Hypertext. HTML was created by Berners-Lee in late 1991 but “HTML 2.0” was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01.

As its name suggests, HTML is a Mark-up Language which means you use HTML to simply “mark-up” a text document with tags that tells a web browser how to structure it to display.

Originally, HTML was develop with the intent of defining the structure of documents like heading, paragraph, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML.

## CSS

Cascading Style Sheet is a style sheet language used for describing the presentation of a document written in a markup language Although most often used to set the visual style of web page and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties .A style sheet consists of a list of *rules*. Each rule or rule- set consists of one or more *selectors*, and a *declaration block*.

## Bootstrap

Bootstrap is a free and open-source, front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only. As of Bootstrap 4, Sass is used instead of less for the style sheets. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code.

**JavaScript**

JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, object-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content. It is used to make WebPages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web browsers support it without the need for plug-ins by means of a built-in JavaScript engine. Each of the many JavaScript engines represent a different implementation of JavaScript, all based on the ECMAScript specification, with some engines not supporting the spectrum fully, and with many engines supporting additional features beyond ECMA.

## MySQL

MySQL is an open source RDBMS that relies on SQL for processing the data in database. MySQL provides APIs for the languages like C, C++, Eiffel, JAVA, Perl, PHP and Python. The code written in MySQL is not case sensitive. MySQL is most commonly used for web applications and for embedded applications and has become a popular alternative to proprietary database system because of its speed and reliability.

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web based software applications. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses.

The following is a list of the main operations that can be formulated with SQL:

* + - * Creating new databases
      * Deleting a database
      * Creating new tables in a database
      * Deleting tables from a database
      * creating and removing users (database access control) executing queries against a database , retrieving data from a database , inserting records in a database , updating records in a database , deleting records from a database

# Advantages of MySQL

* It’s significantly less expensive than most other database options on the market
* It’s compatible with virtually every operating system, and is more or less an industry standard
* MySQL is very easy to install, and thanks to a bevy of third-party tools that can be added to the database, setting up an implementation is a relatively simple task

## PYTHON FLASK

**Flask** is a micro web framework written in Python based on Werkzeug, Jinja2 and inspired by Sinatra Ruby framework, available under BSD license. Flask was created by Armin Ronacher of Pocoo, an international group of Python enthusiasts formed in 2004. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Flask supports extensions that can add application features as if they were implemented in Flask itself.Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program. Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

## Advantages of Python Flask

* Flask has a lightweight and modular design, so it easy to transform it to the web framework you need with a few extensions without weighing it down
* ORM-agnostic: you can plug in your favourite ORM e.g. [SQLAlchemy.](https://quintagroup.com/cms/python/sqlalchemy)
* Basic foundation API is nicely shaped and coherent.
* HTTP request handling functionality
* High Flexibility (The configuration is even more flexible than that of Django, giving you plenty of solution for every production need).

## CHAPTER 3 SYSTEM DESIGN

* 1. **INPUT DESIGN**

Input Design is the process of converting a user-oriented description of the input into a computer-based system. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. Input Design is the first phase in the system design. Input designing is to converting the user oriented information to the computer oriented form. The input data items are grouped and analyzed to find out whether the proposed system can be developed from the user input. The system is developed using various screens formats.

## Adding customer details

This form contains a list of fields that is used to input data about customer. To add a new customer it is required to fill the given form which contain such as name of the customer, mailid, password, phone number and address of the customers are taken as inputs from the customer and added into the application by the customers and then submit it.

## Adding product details

This form is to add the products and offers into the application. The details such as product name, product description, product category, product price, product making price and product images are taken as input from admin and added into the application.

## Adding offer details

As like the form for product details, the offer details contain minimum value, maximum value and discount percentages which are taken as input added into the application by the admin.

## DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The main objectives of designing a database are,

* + - Data Integration
    - Data Integrity
    - Data Independence

## TABLE DESIGN

**Table 3.1: Admin**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| admin\_id | Int | 11 | Primary key |
| admin\_name | Varchar | 50 | Not null |
| admin\_mailid | Varchar | 30 | Not null |
| admin\_password | Varchar | 20 | Not null |
| admin\_phno | Bigint | 12 | Not null |

**Table 3.2: User**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **SIZE** | **CONSTRAINT** |
| customer\_id | Int | 11 | Primary key |
| customer\_name | Varchar | 50 | Not Null |
| Password | Varchar | 20 | Not Null |
| Mailid | Varchar | 30 | Not Null |
| Phoneno | Bigint | 15 | Not Null |
| Address | Varchar | 250 | Not Null |
| District | Varchar | 50 | Not Null |
| Pincode | Int | 6 | Not Null |

**Table 3.3: Product**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| product\_id | Varchar | 20 | Primary Key |
| product\_name | Varchar | 50 | Not Null |
| product\_description | Text | - | Not Null |
| product\_type | Varchar | 50 | Not Null |
| product\_ornament | Varchar | 30 | Not Null |
| product\_weight | Float | 5,2 | Not Null |
| product\_purity | Float | 5,2 | Not Null |
| product\_makingprice | Double | 10,2 | Not Null |
| product\_totalprice | Double | 10,2 | Not Null |
| product\_image1 | Long blob | - | Not Null |

**Table 3.4: Offer**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| offer\_id | Int | 20 | Primary Key |
| min\_value | Int | 20 | Not Null |
| max\_value | Int | 20 | Not Null |
| Discount | Int | 11 | Not Null |

**Table 3.5: Sales**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| sales\_id | Varchar | 20 | Primary Key |
| product\_id | Varchar | 20 | Foreign Key |
| product\_stock | Int | 11 | Not Null |
| product\_sold | Int | 11 | Not Null |
| product\_saleprice | Double | 10,2 | Not Null |
| sales\_date | Date | - | Not Null |

**Table 3.6: Cart**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| cart\_id | Varchar | 20 | Primary key |
| customer\_id | Varchar | 20 | Foreign Key |
| product\_id | Varchar | 20 | Foreign Key |
| Price | Double | 10,2 | Not Null |
| Quantity | Int | 11 | Not Null |
| total\_price | Double | 10,2 | Not Null |

**Table 3.7: Order**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| order\_id | Int | 11 | Primary key |
| cart\_id | Int | 11 | Foreign Key |
| customer\_id | Int | 20 | Foreign Key |
| order\_date | Date | - | Not Null |

**Table 3.8: Placeorder**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| order\_id | Int | 11 | Foreign Key |
| customer\_id | Int | 11 | Foreign Key |
| Name | Varchar | 50 | Not Null |
| phonenumber | Bigint | 12 | Not Null |
| Address | Varchar | 250 | Not Null |
| City | Varchar | 50 | Not Null |
| State | Varchar | 50 | Not Null |
| Pincode | Int | 6 | Not Null |
| Country | Varchar | 50 | Not Null |

**Table 3.9: Invoice**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| Invoice\_no | Int | 11 | Primary key |
| order\_id | Varchar | 20 | Foreign Key |
| customer\_id | Varchar | 50 | Foreign Key |
| offer\_id | Int | 50 | Foreign Key |
| date | Date | - | Not Null |
| items | Int | 10 | Not Null |
| totalamount | Decimal | 10,2 | Not Null |

**Table 3.10: Feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **SIZE** | **CONSTRAINT** |
| customer\_id | Int | 20 | Foreign Key |
| product\_id | Varchar | 11 | Foreign Key |
| quality\_rating | Int | 6 | Not Null |
| design\_rating | Int | 6 | Not Null |
| price\_rating | Int | 6 | Not Null |
| discount\_rating | Int | 6 | Not Null |
| overall\_rating | Int | 6 | Not Null |
| feedback | Varchar | 100 | Not Null |
| date | Date | - | Not Null |

**3.3 MODULE DESCRIPTION**

The main modules exists in this project are listed below

* + - Registration
    - Product details
    - Order placing and Offer Details
    - Sales Details
    - Generating bill

## Registration

The registration module consists of the phase where the customer and admin can register their details. They can login using the mailid and password that they created.

## Product Details

In the product details module the jewellery product details and offer details are added by the admin after verifying the necessary proofs. The product details includes its category and ornament it belongs, purity, making price of the jewellery products and its total prices. The manipulation of the data that has been given by the jewellery shop can only be done by the admin.

## Order placing and offer details

This module helps a customer to order the jewellery products. The users can choose the jewellery product based on the necessity. All the needed basic details will be shown to the customer, in a way the process of decision making becomes much more efficient and convenient. To order a jewel in a website the customer has to sign up and login. But the customers can surf among the various jewellery products that are displayed through the application without logging in. The customer can view the offers that are displayed which have been added by the admin. The customers can post and share their opinions which help other customers to choose what suits best for their need. Through the ratings and feedback given by the customers the promotion strategies can also be improved.

## Sales Details

The sales details contain information about the product such as the stock of the product that are available, customer details , date of the jewellery product purchased and their prices. The manipulation of the data that has been given by the jewellery shop can only be done by the admin.

## Generating bill

In this module the bill will be generated to the customer for the products, they have purchased. The bill contains details about the customer, products, discount price and the date on which they have purchased. Discount amount will be displayed in the bill.

## 3.4 OUTPUT DESIGN

The computerized output is the most important and the direct source of information to the user. The output design defines the output required and the format in which it is to be displayed. The purpose is to produce the required output for the system to reach its success. The outputs are the most important sources of information to the users. Better design should improve the system's relationship with user and also will help in decision making.

## Sales Report

The sales report contains details about the jewellery product that have been sold to the customer. This report is generated from order table. The report is generated between the given time interval.

**Stock Report**

The stock report contains details about the present quantity and quantity of stock that is available after sale for the particular jewellery product code mentioned.

## Offer Report

The offer report contains details about the offers given to the customers. The report is generated between the given time interval. If time interval is not mentioned, it will generate report from the starting date of purchase to the current date.

## CHAPTER 4 SYSTEM TESTING

**SYSTEM TESTING**

Testing is an integral part of any system development life cycle. Software testing includes selecting test data that have more probability of giving errors. System Testing is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective. It is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

The first step in system testing is to develop a plan that tests all aspects of the system. Completeness, correctness, reliability and maintainability of the software are to be tested for the best quality assurance that the system meets the specification and requirements for its intended use and performance. It is the most useful practical process of executing a program with the implicit intention of finding errors that make the program fails. System testing is done in three phases.

* + - * Unit testing
      * Integration testing
      * Validation testing

**Unit Testing**

Developers typically do the unit testing in order to trace out the bugs in each module of the code. Unit testing is done in parallel with coding. They were also tested for specification to see if they were working as per what the program should do and how it should perform under various conditions. All the forms will be run through the menu to see if the proper sequence is maintained. Whenever an error is encountered, an informative error message will be displayed which informs the user what error is. After completion of form testing, the program will be tested.

In this project all the modules have been tested and verified whether the outputs are produced as expected. All the forms will be run through the menu to see if the proper sequence is maintained. Whenever an error is encountered, an informative error message will be displayed which informs the user what error is. After completion of form testing, the program will be tested.

## Integration Testing

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In this project, all the modules are combined and then the entire project is tested as a whole. The integration testing was performed by integrating one module with another and checkout their functionality and execution. There are no errors and defects found in the system. The system meets all the specified user requirements and found working properly. The new system developed was tested by the acceptance testing method. The user provided test area. Thus the system was successfully tested and it satisfies the user requirements.

During integration testing all the modules like are tested with their integration and that could integrated and manipulated with respect and form in between modules. When modules are executed, backend MySQL database simultaneously verified whether updations of records are made in the database and for avoiding duplicate records.

## Validation Testing

Validation is basically done by the testers during the testing. While validating the product if some deviation is found in the actual result from the expected result then a bug is reported or an incident is raised. Not all incidents are bugs. But all bugs are incidents. Incidents can also be of type ‘Question’ where the functionality is not clear to the tester. Hence, validation helps in unfolding the exact functionality of the features and helps the testers to understand the product in much better way. It helps in making the product more user friendly.

It is said that validation is successful when the software function is systematic manner that can be reasonably accept by the customer. This type of testing is very important because it is the only way to check whether the requirements given by the user have been completely fulfilled. The input given to various form is validated efficiently.

It is tested that the lulu jewellery shopping system provides the necessary details of the products and offers such as the features and the offer price.

## CHAPTER 5

**SYSTEM IMPLEMENTATION**

When the initial design was done for the system, the customers were consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a trail run of the system was carried on. The aim of the trail run was to identify any malfunction of the system.

Different types of data are given as input to the system and the output of the given input is checked. Live data are provided. The inputs for the adding of details for the products and offers are provided.

Each operation is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the program specifications, the computer system and its environment is tested to the satisfaction of the user.

System implementation is made up of many activities. The major activities used in project development are as follows.

**CODING**

Coding is the process, whereby the physical design specifications turned into working computer code. In this application Python Flask is used for coding. This is a stable framework and has a large user community. The controllers, models and views are coded and designed as such the customer experience a smooth working.

**TESTING**

Once the coding process begins, testing for each program module is done in parallel. In this application all the modules are tested in parallel to the development of the each module which helps to make the system error free.

## INSTALLATION

There is no need for the installation process as the application is a web service enabling the customers to make use of the application through a browser online.

## DOCUMENTATION

The necessity of the documentation for the application is negative. The system gives a smooth, efficient and convenient experience to purchase the jewellery. When the customer enters into the application, he can choose any needed operation easily as it will be understandable for all users.

# CHAPTER 6

## CONCLUSION AND FUTURE ENHANCEMENTS

**CONCLUSION**

The LULU JEWELLERY SHOPPING is completed successfully in more reliable and user friendly manner. This system helps the customers by providing an efficient and integrated service of purchasing jewellery with promotion for their purchase. All the details of the Products, Offers and Customer details are stored in a database. The system is tested with various sample inputs and is made to provide an error free and smooth experience to the users.

This system helps the user to shop their desired jewellery products from anywhere at any time thus reducing time, cost and effort factors. The customer sends requests to the server, it processes and extracts the data from database and sends the result back to the customer. All the day-to-day transactions are managed through the web application, and also order, sales and offer reports are generated for administrator.

# FUTURE ENHANCEMENTS

* + Multilingual support can be provided so that it can be understandable by the person of any language
  + Animations can be added to make it more user-friendly and understandable
  + Feedback from customers can be collected as audio file

**APPENDICES**

## APPENDIX I

**SAMLPE CODING**

<! DOCTYPE html>

<html lang="en">

<head>

<Title> Jewellery shopping </title>

s<meta charset="utf-8">

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

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font- awesome/4.7.0/css/font-awesome.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

<style>

body

{

background:url();

background-size: cover;

background-repeat: no-repeat;

font-family: Arial;

font-size:14px;

}

.navbar {

margin-bottom: 0;

border-radius: 0;

padding: 10px;

}

#header h1 a:hover

{

color: #273be2;

text-decoration: none;

}

#top

{

padding:10px 0;

text-align:center;

color:#00ffff;

font-size:25px;

height:20px;

}

form.example input[type=text] {

padding: 10px;

font-size: 17px;

border: 1px solid grey;

float: left;

width: 70%;

background: #f1f1f1;

}

form.example button {

float: left; width:30%;

padding:08px;

font-size: 20px;

background:#2196F3; border: 1px solid grey; border-left: none; cursor: pointer;

}

.container

{

color:#666;

padding:20px;

border-radius:10px;

position: absolute;

margin: auto;

top: 0;

right: 0;

bottom: 0;

left: 0;

width: 350px;

height: 400px;

}

</style>

</head>

<body>

<nav class="navbar navbar-inverse">

<img src="{{ url\_for('static', filename='images/logo.jpeg) }}" align="left" height="50" width="200">

<div class="navbar-brand"> <font color="White"><center>LULU ONLINE JEWELLERY SHOPPING </center></font>

</div>

<div class="container-fluid">

<div class="collapse navbar-collapse" id="myNavbar">

<ul class="nav navbar-nav navbar-right">

<li><a href="register"><span class="glyphicon glyphicon-login"></span>Signup</a></li>

<li><a href="login"><span class="glyphicon glyphicon-log-in"></span>Login</a></li>

</ul>

</div>

</div>

</nav>

<div class="container">

<form action="/register" method="post" autocomplete="off">

<h1><center>REGISTER</center></h1>

<div class="form-group">

<p><span class="glyphicon glyphicon-user"></span> Name:</p>

<input type="text" class="form-control" id="name" name="name" pattern="[A-Za- z]{1,12}" placeholder="Name" required>

</div>

<div class="form-group">

<p><span class="glyphicon glyphicon-lock"></span> Password:</p>

<input type="password" class="form-control" id="password" pattern="(?=.\*\d)(?=.\*[a- z])(?=.\*[A-Z]).{8,}" name="password" placeholder="Password" required>

</div>

<div class="form-group">

<p><span class="glyphicon glyphicon-lock"></span>Conform Password:</p>

<input type="password" class="form-control" id="confirm" name="confirm" placeholder="Conform Password" required>

</div>

<div class="form-group">

<p><span class="glyphicon glyphicon-envelope"></span> Email:</p>

<input type="email" class="form-control" id="mailid" name="mailid" style="text-

transform: lowercase" placeholder="Email Id" required>

</div>

<div class="form-group">

<p><span class="glyphicon glyphicon-phone"></span> Phone Number:</p>

<input type="text" class="form-control" id="phno" pattern="^\d{10}$" title="Please enter 10 digits" name="phno" placeholder="Phone Number" required>

</div>

<div class="form-group">

<p><span class="glyphicon glyphicon-phone"></span> Address:</p>

<input type="text" class="form-control" id="phno" pattern="^\d{10}$" title="Please enter 10 digits" name="phno" placeholder="Address" required>

</div>

<center><button type="submit" class="btn btn-default" name="submit"><u><b>Submit</u></b></button></center>

</form>

</div>

<div class="col-sm-3 text-left">

</div>

<div class="container-fluid text-center">

<div class="col-sm-6 text-center">

<br>

</div>

</div>

<div class="col-sm-3 text-left">

</div>

</body>

</html>

## PYTHON CODE:

import os

from flask import Flask, render\_template, flash, redirect, url\_for, logging, request, session

import mysql.connector

from werkzeug import secure\_filename import datetime

app = Flask( name )

connection = mysql.connector.connect(host='localhost', database='jewellery',

user='root', password='', buffered=True)

@app.route('/register', methods=['GET','POST']) def register():

if request.method == 'POST':

name = request.form.get('name')

password = request.form.get('password')

confirm = request.form.get('confirm')

mailid = request.form.get('mailid')

phno = request.form.get('phno')

address = request.form.get(‘address’)

cur = connection.cursor()

if password == confirm:

cur.execute("INSERT INTO user(customer\_name,password,mailid,phno,address) VALUES(%s,%s,%s,%s)",(name,password,mailid,phno,address))

connection.commit()

cur.close()

flash('You are Registered successfully','success') return redirect(url\_for('login'))

else:

flash('You are password does not match','failed') return render\_template('customer\_register.html') return render\_template('customer\_r