

ONLINE HORTICULTURE SHOP

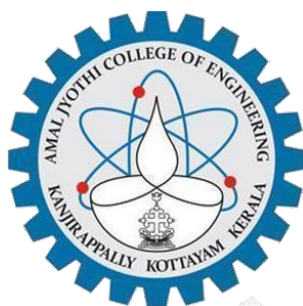
Project Report Submitted By

HIMA M S

Reg. No: AJCE20MCA-2041

In Partial fulfillment for the Award of the Degree Of

**MASTER OF COMPUTER APPLICATIONS (2 year)
(MCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2020-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “ **ONLINE HORTICULTURE SHOP**” is the bonafide work of **HIMA M S (Reg No: AJCE20MCA-2041)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-2022

Ms.Rini Kurian
Internal Guide

Ms. Nimmy Francis
Coordinator

Rev.Fr.Dr. Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report “**ONLINE HORTICULTURE SHOP**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfillment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date: 21/07/2022

KANJIRAPPALLY

HIMA M S

Reg. No: AJCE20MCA-2041

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev. Fr. Dr. Rubin Thottupurathu Jose** and **Ms. Nimmy Francis** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Rini Kurian** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

HIMA M S

ABSTRACT

“ONLINE HORTICULTURE SHOP” is a web application. The central concept of the application is customers to purchase the product virtually based on their wishes. Then the items are shipped to the address submitted by them. The application is designed into four actors: Customers, Admin, Stock Manager, Delivery Boy. First one is customer, customer can view all the products and its details, and wishes to buy the item then they have to purchase after the payment. Each customer has their own accounts containing the personal information like shipping and billing address etc, order details like order status and wishlist's, using a unique order id they can track the order and check the current status. The second one is Admin, admin added and updated the product and its details also add new category and subcategory of products and view the customers information's like their full details and login and logout time. Another actor is Stock Manager, stock manager manages the out of stock products. When a product will show out of stock then add more quantity of that product and they can view the total products and orders also create a financial report. Report generation is one of the main functionality it based on the date of ordering. Last one is Delivery Boy, delivery boy manage the orders. They can view today's orders, pending orders and delivered orders then delivery boy change the status of customers orders. Customers can track the status on their profile. Payment is one of the functionality of this application after the payment user can generate the corresponding bill.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	PROPOSED SYSTEM	6
2.4	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	8
3.1	FEASIBILITY STUDY	9
3.1.1	ECONOMICAL FEASIBILITY	9
3.1.2	TECHNICAL FEASIBILITY	10
3.1.3	BEHAVIORAL FEASIBILITY	10
3.2	SYSTEM SPECIFICATION	11
3.2.1	HARDWARE SPECIFICATION	11
3.2.2	SOFTWARE SPECIFICATION	11
3.3	SOFTWARE DESCRIPTION	11
3.3.1	PHP	11
3.3.2	MYSQL	12
4	SYSTEM DESIGN	14
4.1	INTRODUCTION	15
4.2	UML DIAGRAM	15
4.2.1	USE CASE DIAGRAM	16
4.2.2	SEQUENCE DIAGRAM	18
4.2.3	STATE CHART DIADRAM	24
4.2.4	ACTIVITY DIAGRAM	26

4.2.5	CLASS DIAGRAM	27
4.2.6	OBJECT DIAGRAM	28
4.2.7	COMPONENT DIAGRAM	29
4.2.8	DEPLOYMENT DIAGRAM	30
4.3	USER INTERFACE DESIGN	31
4.4	DATA BASE DESIGN	35
4.4.1	RELATIONAL DATABASE MANAGEMENT SYSTEM	35
4.4.2	NORMALIZATION	36
5	SYSTEM TESTING	43
5.1	INTRODUCTION	44
5.2	TEST PLAN	45
5.2.1	UNIT TESTING	45
5.2.2	INTEGRATION TESTING	46
5.2.3	VALIDATION TESTING	46
5.2.4	OUT PUT TESTING	46
5.2.5	AUTOMATION TESTING	47
5.2.6	SELENIUM TESTING	47
6	IMPLEMENTATION	52
6.1	INTRODUCTION	53
6.2	IMPLEMENTATION PROCEDURE	53
6.2.1	USER TRAINING	54
6.2.2	TRAINING ON APPLICATION SOFTWARE	54
6.2.3	SYSTEM MAINTENANCE	54
7	CONCLUSION & FUTURE SCOPE	56
7.1	CONCLUSION	57
7.2	FUTURE SCOPE	57
8	BIBLIOGRAPHY	58
9	APPENDIX	60
9.1	SAMPLE CODE	61
9.2	SCREENSHOTS	72

List of Abbreviations

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

“ONLINE HORTICULTURE SHOP ”The central concept of the application to allow the customer to shop product virtually and pay the corresponding amount. Then the items are shipped to the address submitted by them. The application is designed int four login , first is for the customer who wish to buy the item .Customer can view all product with out login but they wish to buy that particular item then show the login option after the order customer can pay the corresponding amount of ordered item. Second is for the admin who maintains and updates the information to the product and customer. Admin add new category and subcategories of products. Another actor is Stock Manager, stock manager manages the out of stock products. when a product will show out of stock then add more quantity of that product and they can and view the total products and orders also create a financial report. Report generation is one of the main functionality it based on the date of ordering. Last one is Delivery Boy, delivery boy manage the orders. They can view today's orders, pending orders and delivered orders then delivery boy change the status of customers orders. customers can track the status on their profile.

1.2 PROJECT SPECIFICATION

The system includes 4 modules. They are:

1. Admin Module

Admin must have a login into this system . Admin can view the user details and add new products also add new categories and subcategory products.

2. Customer Module

Customer can view all the plants and they can order the plants based on their wish then it will delivered the corresponding address after the payment. Each customer have their on profile it containing the personal information and order details. customer can track the order status using the order id.

3. Stock Manager Module

Stock manager manage the out of stock products and create reports based on product ordering date. Stock manager can view the all products and all orders. when a product show out of stock then stock manager add the product. also view the product review.

4. Delivery Boy Module

Delivery boy can view today's order ,pending order ,delivered order and update the status like in process delivered or not yet proceed.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decision variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal. Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

Existing system is not automated system. Customer can go to the shop, and purchase the product directly. In proposed system customer can purchases the product through online, and product to get their on places and customer to pay the money only after the delivery

It is necessary to modify the existing system in order to include additional information and it will fully automated. and make the system efficient, flexible and secure. Using the new system customers can viewing the profile and view the order details.

2.3 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for business growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the customers, stoker for manage stock, delivery boy for delivery services . The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Our website is multifunctional which includes customer introduction, admin introduction, product details etc.

2.4 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:.

○ Ensure data precision: -

The proposed system eliminates the manual errors while entering the details of the users during the registration.

○ Enhanced safety:-

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

○ Better service: -

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

To ascertain if the project will, after completion, achieve the goals of the organization in relation to the work, effort, and time put into it, a feasibility study is carried out. The developer can forecast the project's usefulness and potential future thanks to a feasibility study. The viability of the system concept, which includes its impact on the organization, capacity to satisfy user needs, and efficient use of resources, serves as the foundation for a feasibility study. As a result, a feasibility evaluation is frequently performed before a new application is approved for development.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including such as Technical, Economic and Operational feasibility. It has the following characteristics:

3.1.1 Economical Feasibility

Cost and benefit analyses are required to support the emerging system. criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables. Some significant financial concerns raised during the initial study include the following.

- The costs conduct a thorough system investigation.
- The cost of the hardware and software.
- The benefits manifest themselves as lower costs or less costly errors.

There are no manual costs involved with the suggested system because it was developed as part of a project. Additionally, the fact that all of the resources are already at hand indicates that the system may be developed affordably. The cost of the project was divided according to the system used, its development cost, and the cost of hosting the project. According to all the calculations the project was developed at a low cost. As it is completely developed using open-source software.

3.1.2 Technical Feasibility

The system must first undergo a technical evaluation. The assessment of its viability must be built upon an overview design of the system's requirements in terms of input, output, programmers, and procedures. The inquiry must next advise the kind of equipment, necessary procedure for constructing the system, and means of operating the system once it has been designed after having identified an outline system. The following technical difficulties came up throughout the investigation:

- Does the proposed technology function with existing technology?
- If the system has been developed, can it expand?

The project should be built in such a way that the required performance and functionality are met within the limits. The system may still be used even though the technology may become outdated after a while because a newer version of the same software still works with an earlier version. Therefore, this project only has a few limitations. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The system was developed with PHP for the front end and a MySQL server for the back end; the project can be finished technically. The system was effectiveness terms of processor performance.

3.1.3 Behavioral Feasibility

The proposed system included questions?

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All Behavioral aspects are considered carefully and conclude that the project is behaviorally feasible. Horticulture Shop, GUI is simple so that users can easily use it. Horticulture shop is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	- HTML, CSS
Backend	- MYSQL
Client on PC	- Windows 7 and above.
Technologies used	- JS, HTML5, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language used for both web development and generalpurpose programming. PHP is now used by 2.1 million web servers and more than 244 million webpages. The PHP group now produces the reference implementation of PHP, which was first developed by Rasmus Lerdorf in 1995. PHP, a recursive acronym that was once meant for personal pages, now stands for PHP-hypertext preprocessor. A web server's PHP processor module interprets PHP code to produce the final web page. Instead of calling an external file, PHP commands can be directly put into an HTML source file to handle data. Additionally, because of restrictions on the usage of the name PHP under the GNU General Public License (GPL), it has evolved to now give a command-line interface and be used in standalone programs. Most web servers support the free deployment of PHP, which is also available as a standalone shell on practically all platforms and operating systems.

3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

A planned collection of data is called a database. Anything might be it, including a straightforward shopping list, a photo gallery, or the enormous amount of data in a company network. Data included in a computer database must be added to, accessed, and processed using a database management system, such as MySQL Server. Because computers are so good at processing enormous amounts of data, database management systems whether employed as standalone programs or as a component of other applications are crucial to computing.

- **MySQL databases are relational.**

Instead of placing all the data in one huge warehouse, a relational database keeps the data in individual tables. Physical files that are designed for speed contain the database structures. A flexible programming environment is offered by the logical model, which includes objects like databases, tables, views, rows, and columns. One-to-one, one-to-many, unique, mandatory, optional, and "pointers" between distinct tables are just a few examples of the rules you may make to control the relationships between various data fields. Since a well-designed database upholds these constraints, your application won't ever run into inconsistent, duplicate, orphan, out-of-date, or missing data. The prefix "SQL" in MySQL stands for "Structured Query Language." SQL is the most widely used standard language for database access. Depending on your programming environment, you might openly enter SQL (for example, to generate reports), embed SQL statements within other languages' code, instead, employ a language-specific API that hides the SQL coding. By way of the ANSI/ISO SQL Standard, SQL is defined. Since its inception in 1986, the SQL standard has undergone multiple revisions. The 1992 standard is referred to in this document as "SQL92," the 1999 standard is referred to as "SQL," and the most recent version of the standard is referred to as "SQL: 2003." "The SQL standard" refers to the SQL.

- **MySQL software is Open Source.**

Anyone can use and modify the software because it is open source. The MySQL software is available for free download and usage any person online. You have the right to look at the source code and make any necessary changes. The GPL (GNU General Public License), which specifies what you may and cannot do with the programme under particular circumstances, is followed by the MySQL software. You can purchase a commercially licensed version from us if the GPL makes or if you need to integrate MySQL code into a for-profit application. For further details, see the MySQL Licensing Overview.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

A multi-threaded SQL server, several client programmes and libraries, management tools, and a wide variety of application programming interfaces (APIs) make up the client/server system known as the MySQL Database Software. Additionally, we provide MySQL Server as a built-in multi-threaded library that you can link into your programme to create a standalone solution that is more manageable, speedy, and simple to use.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Any engineered system or product's development process begins with design. A creative process is designed. The secret to an efficient system is a decent design. "Design" is the process of using many approaches and concepts to thoroughly outline a process or a system so that it can be physically implemented. One way to describe it is as the process of using different methodologies and concepts to specify a device, a process, or a system in enough detail to allow for its physical reality. Regardless of the development paradigm that is employed, software design forms the technical core of the software engineering process. The architectural detail needed to construct a system or product is developed through the system design. This program has also through the best possible design phase, fine-tuned all efficiency, performance, and accuracy levels, as in the case of any systematic technique. During the design stage, a user-oriented document is transformed into a document for programmers or database employees. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

Software system artefact can be specified, visualized, built, and documented using the UML standard language. The Object Management Group (OMG) is the organization that developed UML, and the OMG received a draught of the UML 1.0 definition in January 1997. Unified Modeling Language is known as UML. Compared to other popular programming languages like C++, Java, COBOL, etc., UML is unique. A visual language called UML is used to create software blueprints. A general-purpose visual modeling language for software system visualization, specification, construction, and documentation is what UML is known as. UML is not just used to represent software systems, despite the fact that this is its most common application. It is also used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to generate code using UML diagrams in a variety of languages. The analysis and design of object-oriented systems are directly related to UML. It has been so well standardized that OMG now recognizes UML as a standard. A complete UML diagram, which depicts a system, is created using all the elements and linkages. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. The following eight diagrams are part of UML.

- Use case diagram
- Sequence diagram
- State chart diagram
- Activity diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

4.2.1 Use Case Diagram

An illustration of the interactions between system components is a use case diagram. A use case is a method for locating, defining, and arranging system needs. The word "system" in this context refers to a thing that is being built or operated, such as a website for mail-order service and product sales. Use case diagrams are used in UML (Unified Modeling Language), a standard language for modeling actual items and systems. Planning general requirements, validating hardware designs, testing and debugging software products while they are still in development, creating online help resources, and finishing customer support-focused tasks are a few examples of system objectives. For instance, customer support, item ordering, catalog updating, and payment processing are examples of use cases in a setting of product sales. A use case diagram consists of four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.



4.2.2 Sequence Diagram

A sequence diagram essentially shows how things interact with one another sequentially, or the order in which these interactions occur. A sequence diagram can also be referred to as event diagrams or event scenarios. Sequence diagrams show the actions taken by the components of a system in chronological order. Business people and software engineers frequently use these diagrams to record and comprehend the requirements for new and current systems.

Sequence Diagram Notations –

i. Actors – In a UML diagram, an actor represents a particular kind of role that interacts with the system and its objects. An actor is always beyond the purview of the system that we want to use the UML diagram to represent. We employ actors to portray a variety of roles, including those of human users and other outside subjects. In a UML diagram, an actor is represented using a stick person notation. In a sequence diagram, there might be several actors.

ii. Lifelines – A lifeline is a named element in a sequence diagram that represents an individual participant. So, in a sequence diagram, each incident is represented by a lifeline. A sequence diagram's lifeline elements are at the top.

iii .Messages – Messages are used to show how objects communicate with one another. The messages are displayed on the lifeline in chronological sequence. Arrows are how messages are represented. A sequence diagram's main components are lifelines and messages.

Messages can be broadly classified into the following categories:

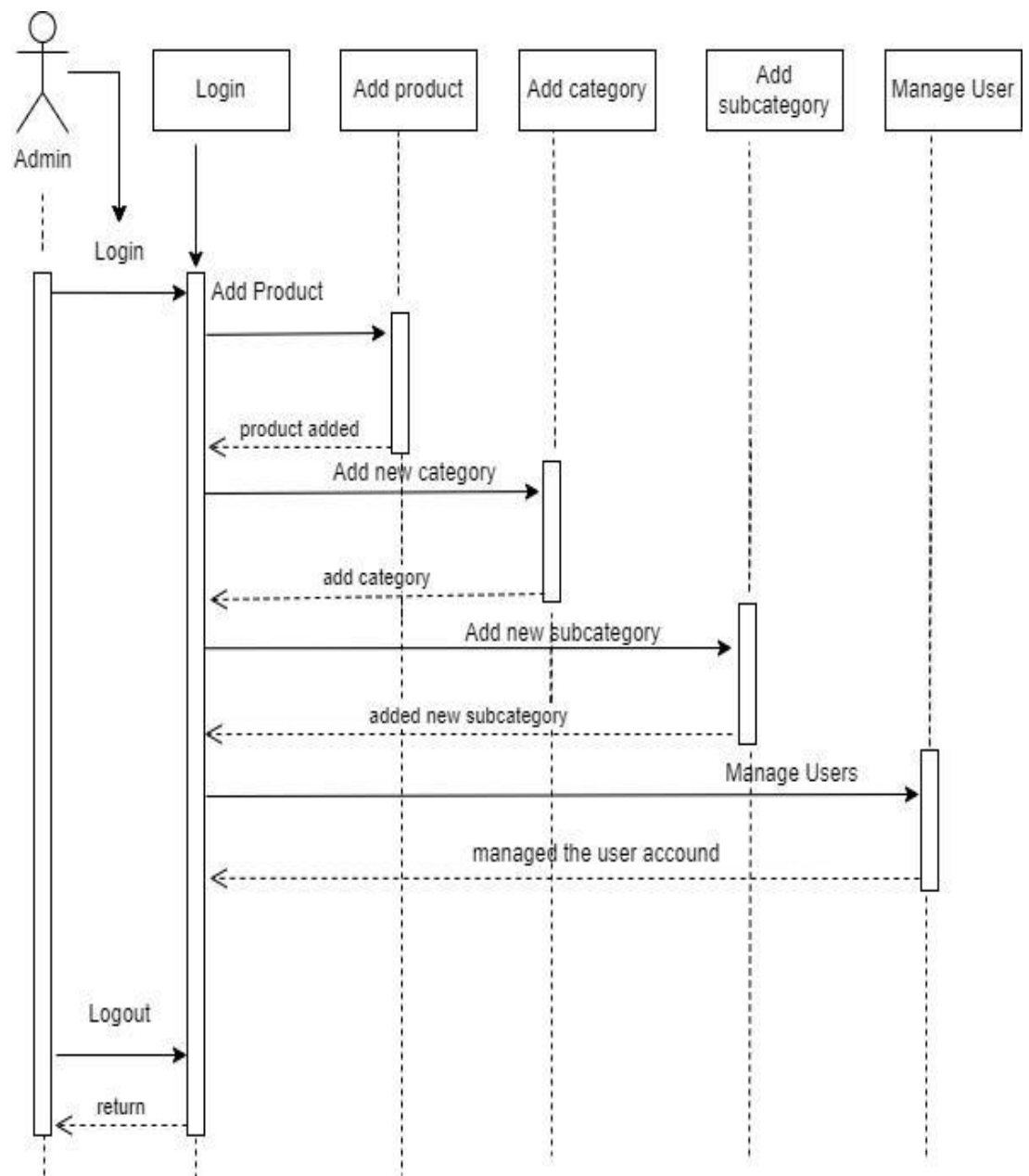
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message

iv. Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

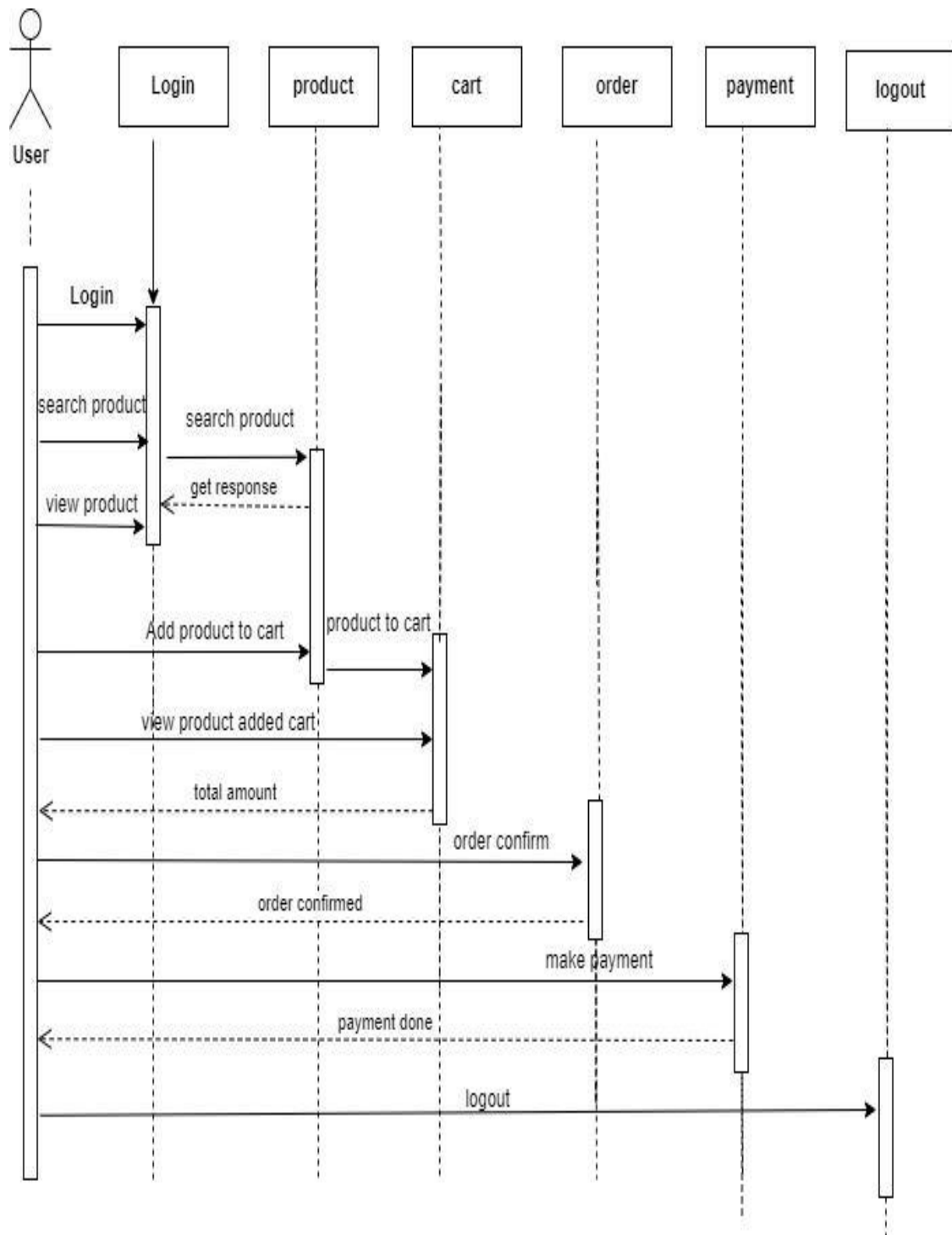
Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.

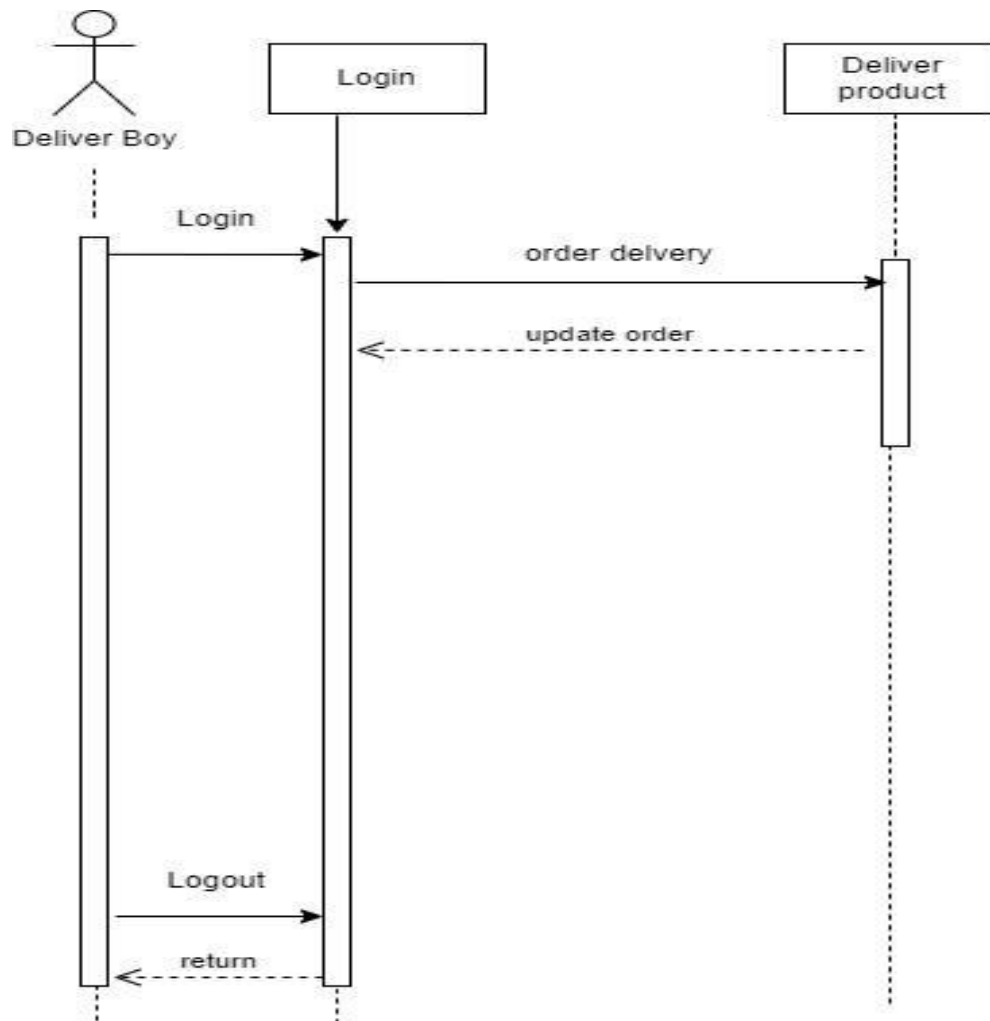
Admin:



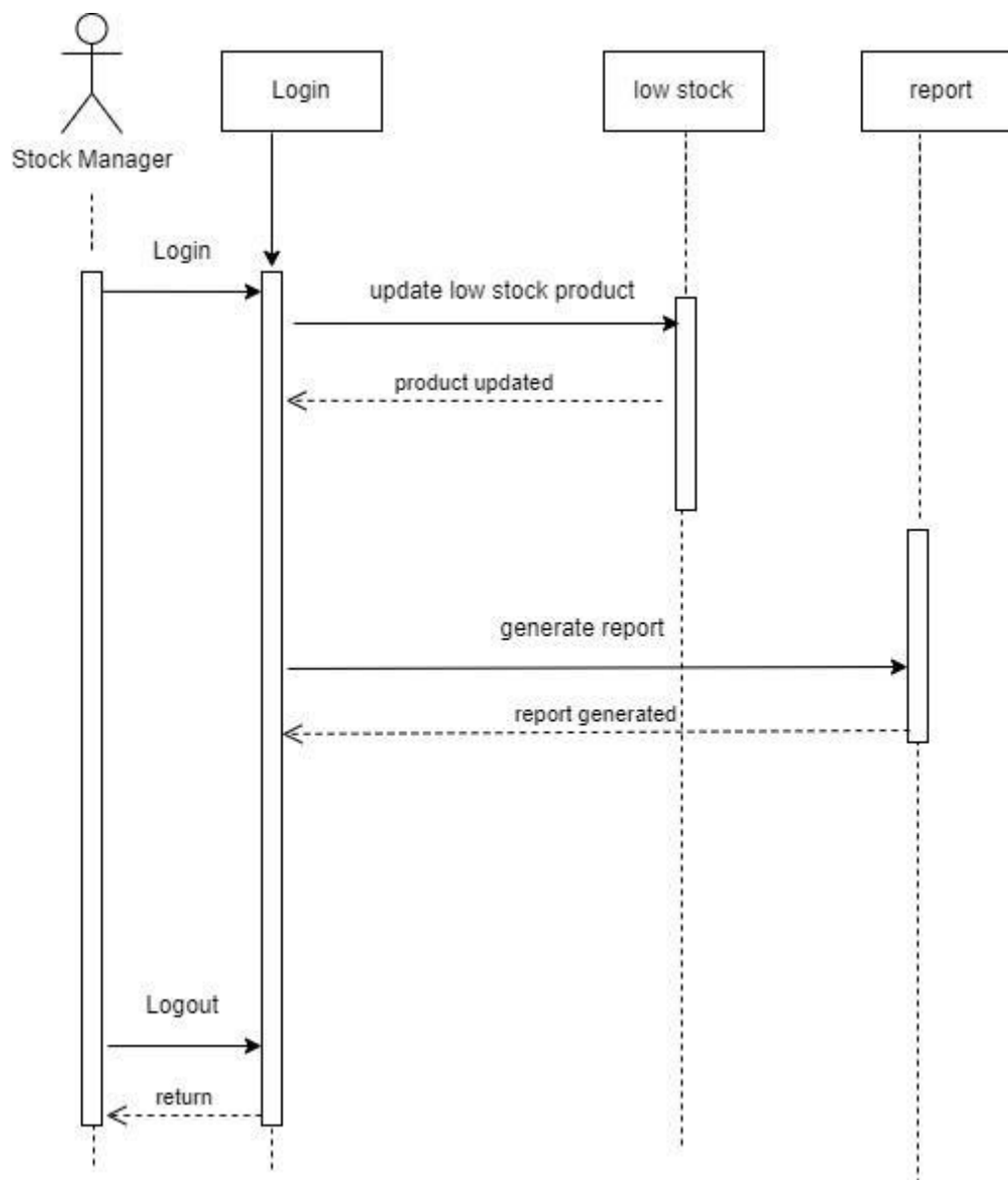
User:



Delivery boy:



Stock Manager:

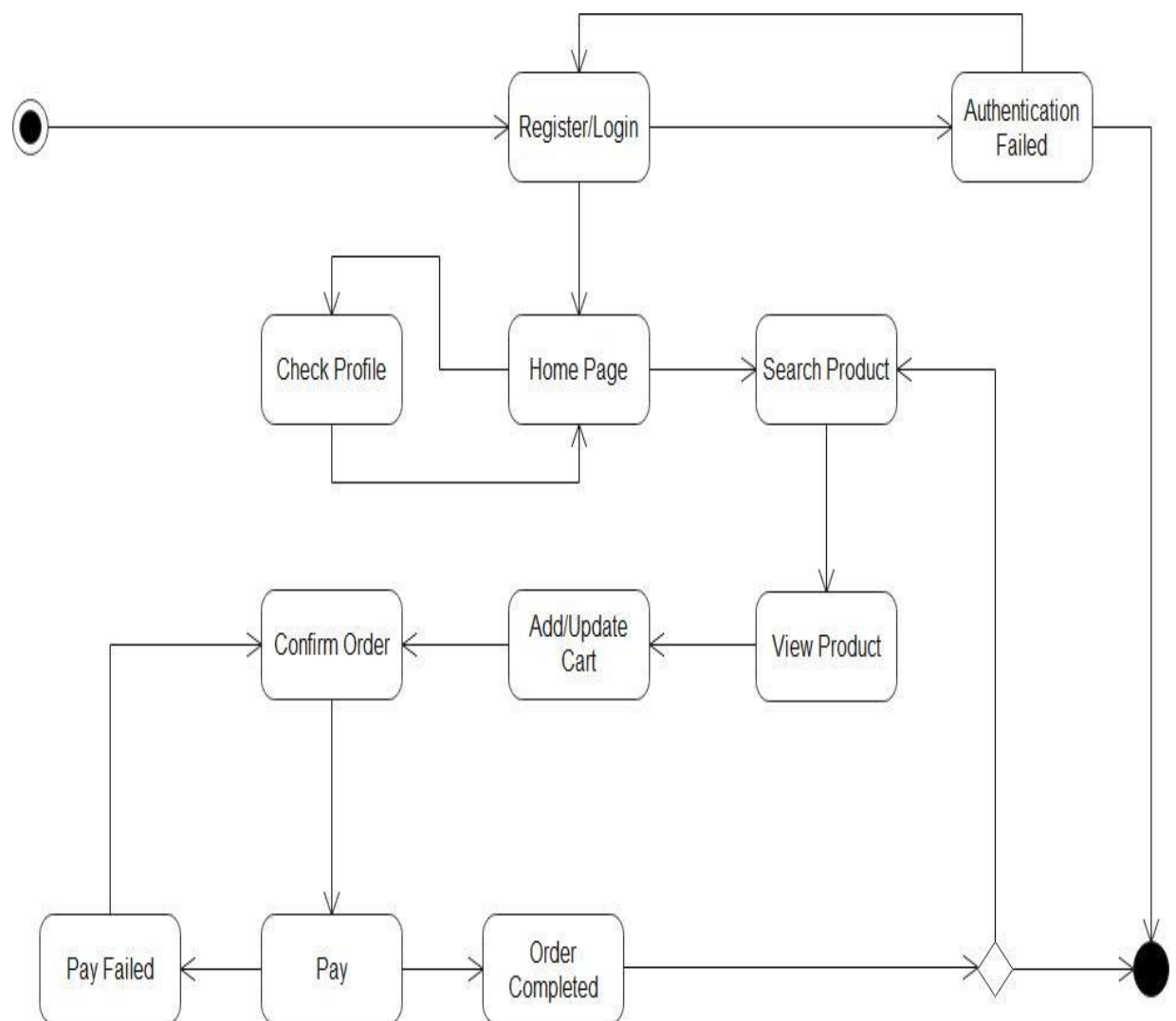


4.2.3 State Chart Diagram

It describes various system components' statuses. The states are unique to a particular system object or component. A state machine is described in a Statechart diagram. A state machine is a device that creates various states for an object and controls these states through external or internal events. Throughout an object's existence, they define several states, and these states are altered by events. The reactive systems can be model led with statechart diagrams. A system that reacts to internal or external events is known as a reactive system. The transfer of control from one state to another is depicted in a state chart diagram. States are described as a situation in which an object is present and changes in response to an event. The main goal of a state chart diagram is to model an object's lifetime from creation to destruction. For both forward and backward engineering of a system, state chart diagrams are employed. But modelling the reactive system is the fundamental objective.

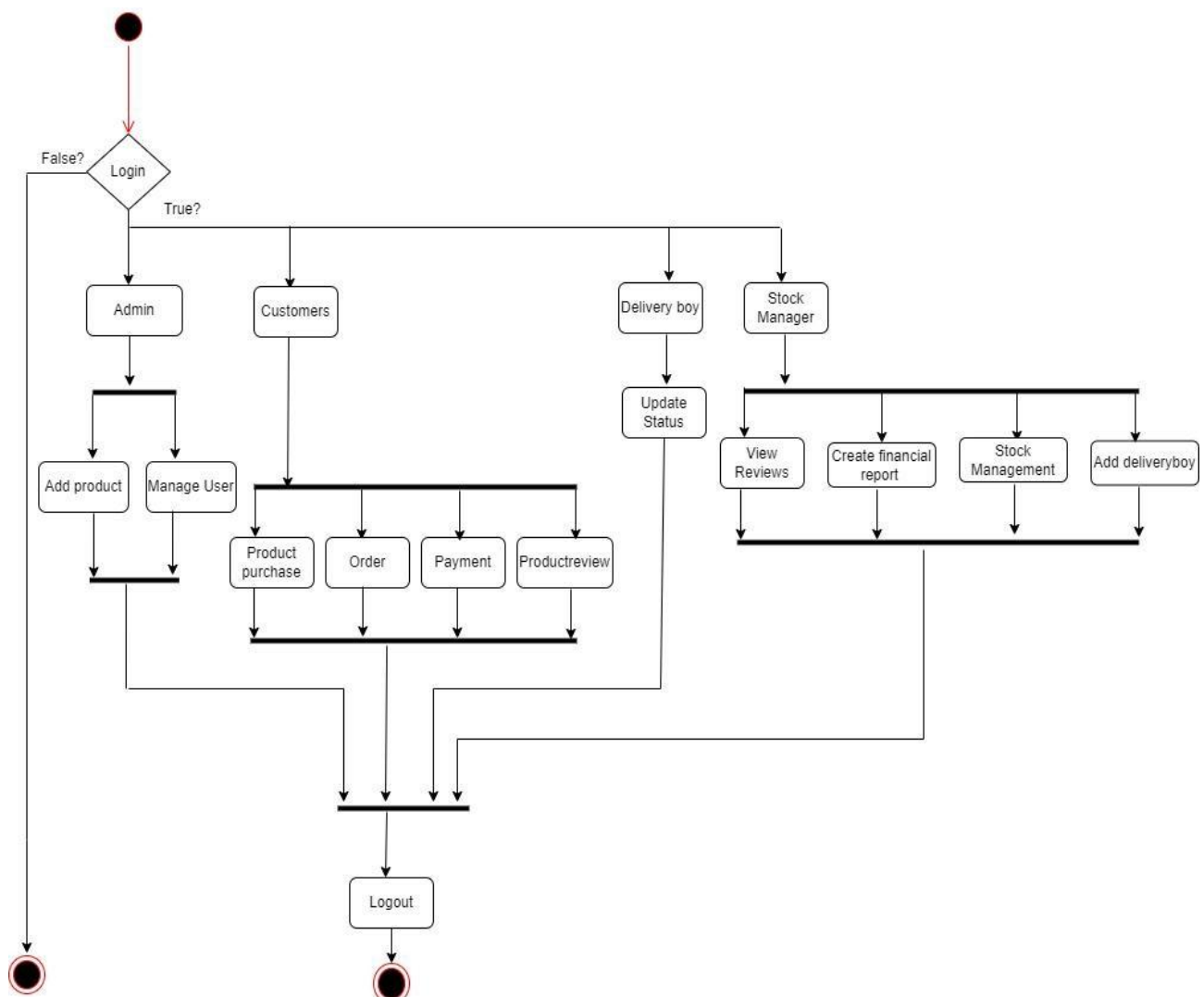
Following are the main purposes of using State chart diagrams –

- To model the dynamic aspect of a system.
- To model the lifetime of a reactive system.
- To describe different states of an object during its lifetime.
- Define a state machine to model the states of an object.



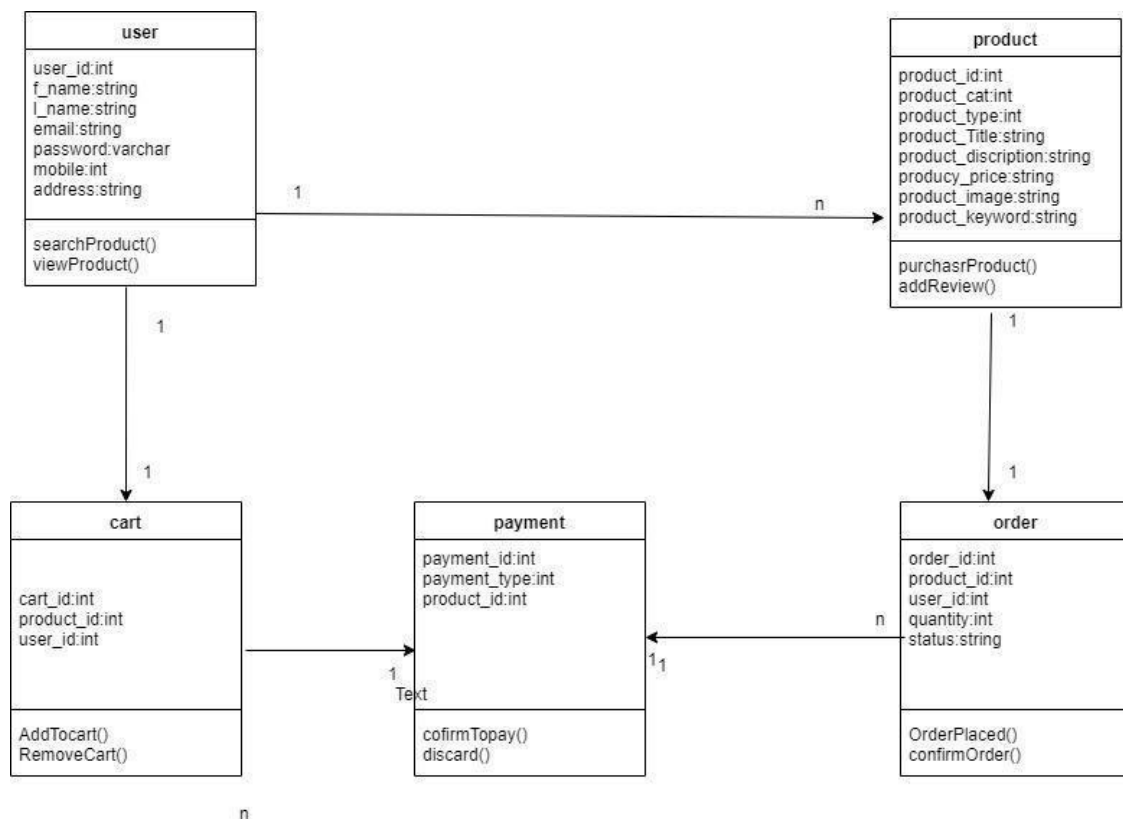
4.2.4 Activity Diagram

The activity diagram describes the system's dynamic properties. A flowchart that illustrates how one activity leads to another is an activity diagram. A system operation could be used to describe the action. In the control flow, one action leads to the next. This flow may be concurrent, parallel, or branching. Many features, including fork, join, and others, are used in activity diagrams to handle various sorts of flow control. Activity diagrams are used to build the executable system utilizing forward and reverse engineering approaches, as well as to visualize the dynamic nature of a system. The message portion is the only item the activity diagram is missing. No message flow from one activity to another is shown. Occasionally, an activity diagram is used in place of a flowchart. The diagrams are not flowcharts, despite their appearance.



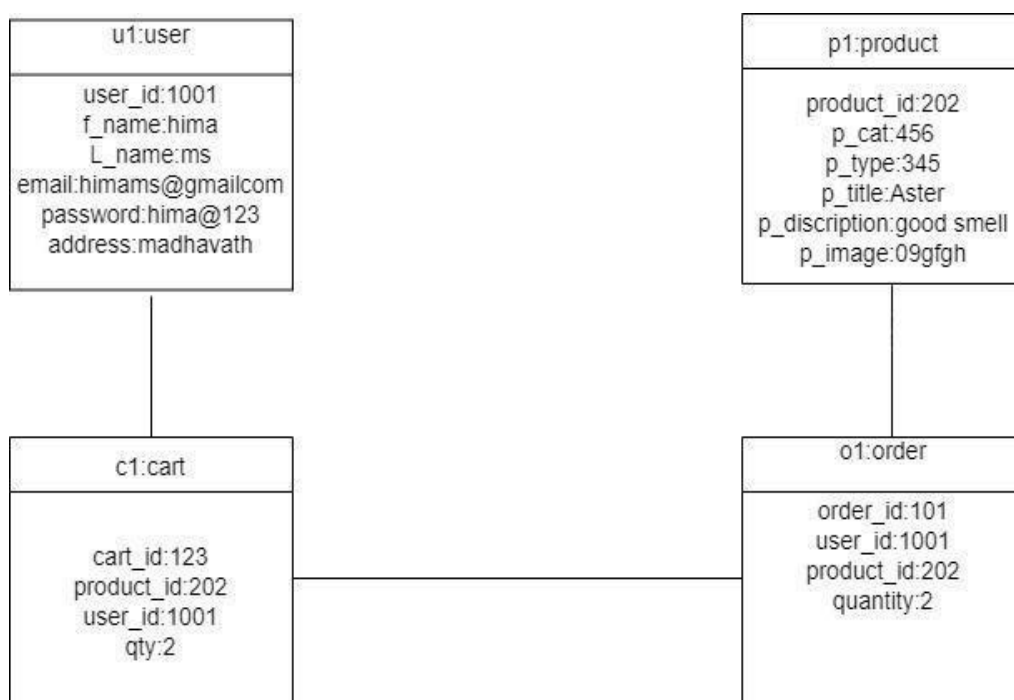
4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object- oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



4.2.6 Object Diagram

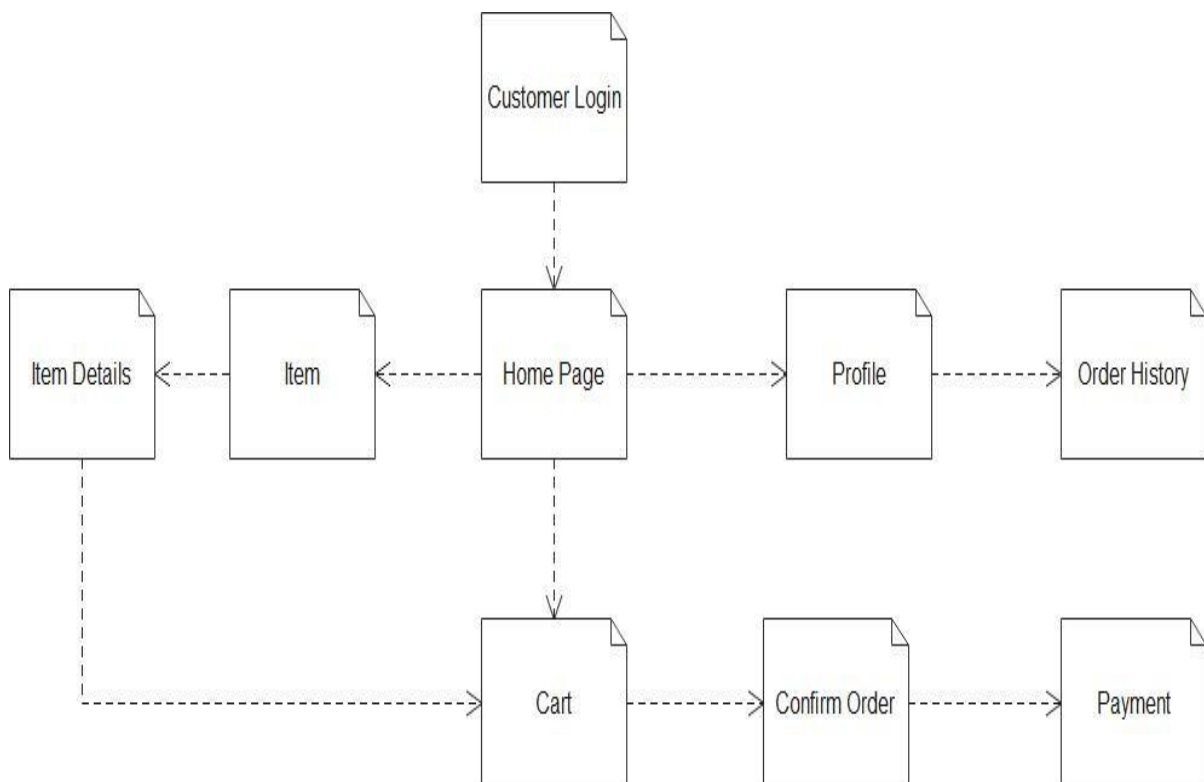
Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.



4.2.7 Component Diagram

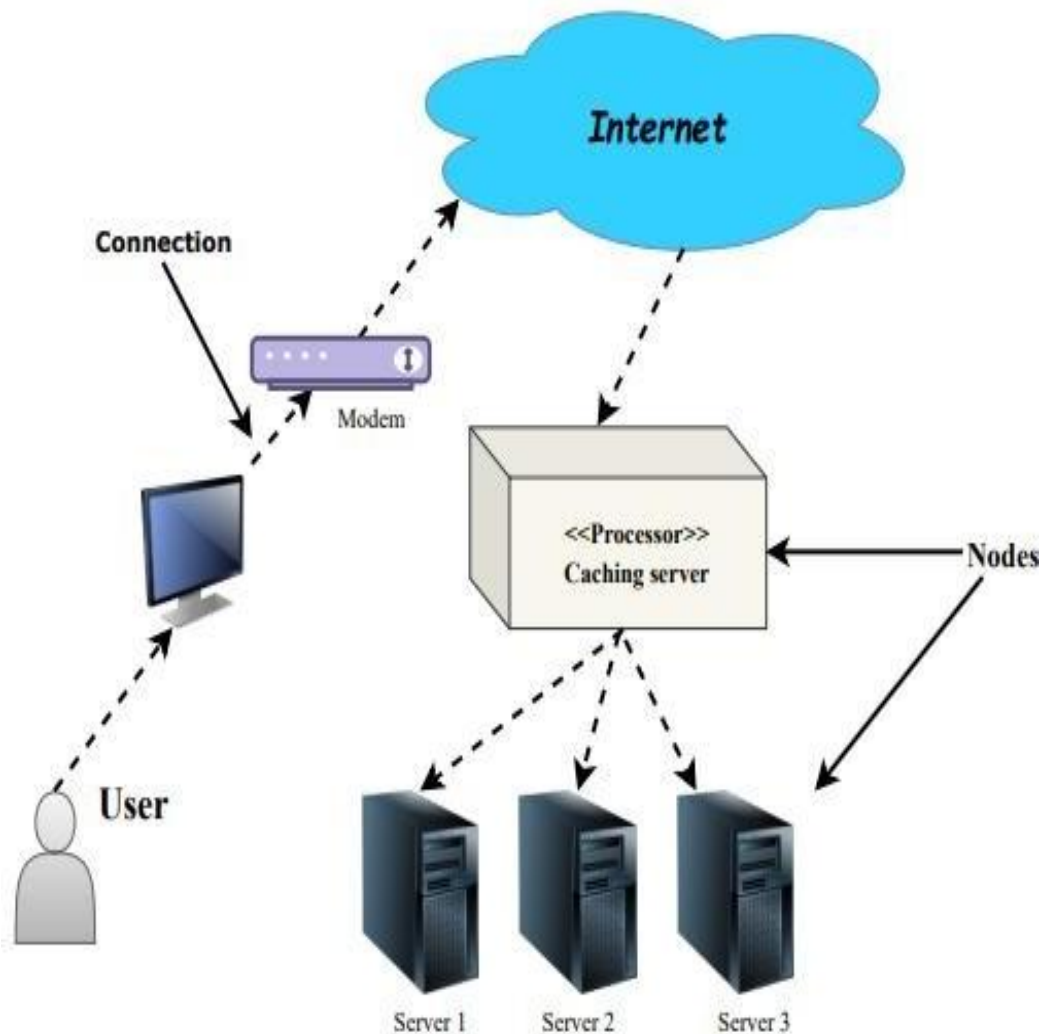
A specific type of UML diagram is a component diagram. In addition, the goal is distinct from the previous diagrams mentioned. Instead, than describing the system's functioning, it discusses the parts that go into creating it. Therefore, from that perspective, component diagrams are utilized to represent the actual physical parts of a system. These parts include files, libraries, and packages, among others. Another way to think of component diagrams is as a static implementation perspective of a system. Static implementation depicts how the components are arranged at a specific time. A group of diagrams are utilized to portray the overall system because a single component diagram is unable to do so. The component diagram's goal can be summed up as:

- Identify the parts of a system visually.
- Utilize both forward and reverse engineering to build an executable.



4.2.8 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.



4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Customer Registration

SIGN IN

Hello, Welcome to your account.

You have successfully logout

Email Address *

Password *

[Forgot your Password?](#)

CREATE A NEW ACCOUNT

Create your own Shopping account.

Full Name *

Email Address *

Email already exists .

Contact No. *

Password *

Confirm Password. *

Product details



Elephant bush, Portulacaria afra, Jade plant (Green) - Succulent Plant

(0 Reviews)

AVAILABILITY : In Stock

PRODUCT TYPE : Indoor

SHIPPING CHARGE : 40

Rs. 220



Payment

Dush 2 Shine

Add more colors to your day

₹ 260

English ▾

+917510516366

|

2018himams@gm...

PREFERRED PAYMENT METHODS

UPI - PhonePe

>

UPI - PayTM

>

CARDS, UPI & MORE

Card

Visa, MasterCard, RuPay, and Maestro

UPI / QR

Stock Management

Total Product21

Total Orders6

Low Stock1

18

Monday 18, 2022

Add DELIVERY BOY

Calendar

July 2022

today < >

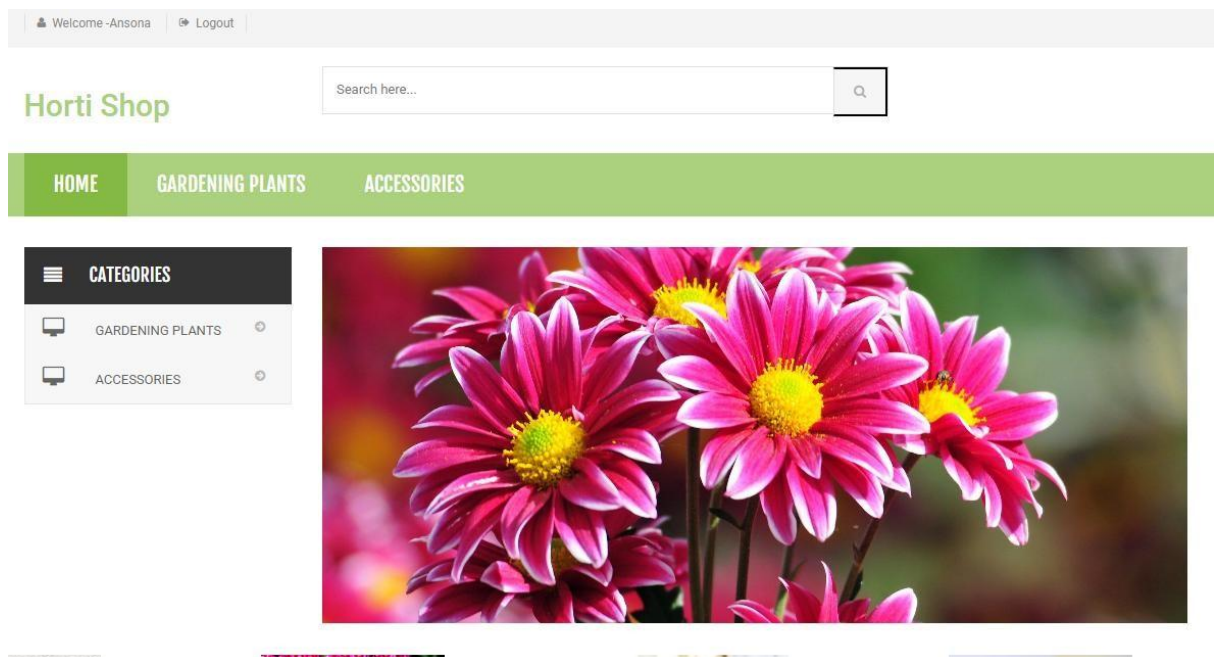
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Amal Jyothi College of Engineering, Kanjirappally

Department of Computer Applications

4.3.2 OUTPUT DESIGN

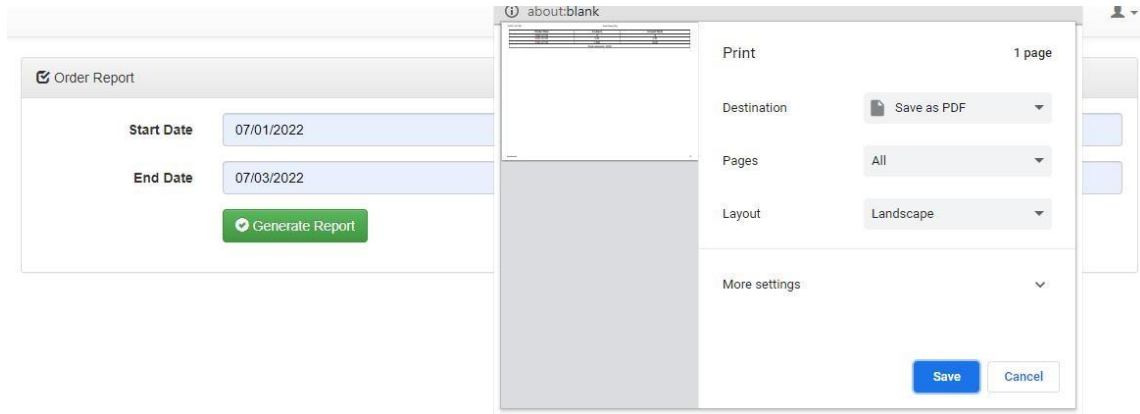
User Account



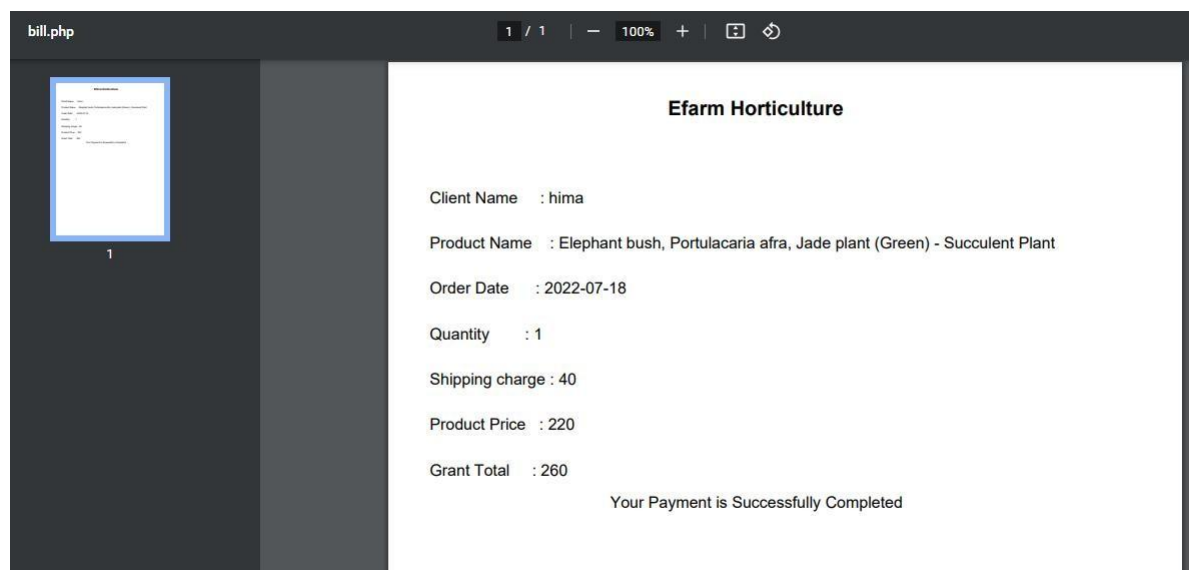
Cart

	LUCK JADE PLANTS IN CERAMIC POTS ★★★★★☆ (0 Reviews)	1	RS 300.00	RS 40.00	340.00
	HANDI SHAPE ROUND CERAMIC POTS ★★★★★☆ (0 Reviews)	1	RS 0.00	RS 40.00	40.00

Report Generation



Bill Generation



4.4 DATABASE DESIGN

A database is a structured system with the capacity to store information and allows users to retrieve stored information quickly and effectively. Any database's primary goal is its data, which demands protection. There are two stages to the database design process. The user needs are obtained in the first step, and a database is created to as clearly as possible meet these criteria. This process, known as information level design, is carried out independently of all DBMS. The design for the specific DBMS that will be used to construct the system in issue is converted from an information level design to a design in the second stage. Physical Level Design is the stage where the characteristics of the particular DBMS that will be used are discussed. Parallel to the system design is a database design. The database's data arrangement aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.4.1 Relational Database Management System

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A domain D is a set of atomic values. A common method of defining a domain is by selecting a data type from which the domain's data values are derived. It is also useful to give the domain a name in order to make the values of the domain simpler to comprehend. A relation's values are all atomic and inseparable from one another.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.4.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. A technique called normalization divides superfluous fields and divides a huge table into smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Normal form in data modelling use two concepts, keys and relationships. A row in a table is uniquely identified by a key. The two different types of keys are primary keys and foreign keys. To identify between records from the same table, a primary key is an element, or group of elements, in the table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been Normalized.

It means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalization prevents data redundancy, which puts a heavy strain on the computer's resources. These consist of

- Normalize the data.
- Choose proper names for the tables and columns.
- Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non- primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN**Table No 01 : admin****Description : To store the admin details****Primary key :id**

Field Name	Type	Size	Description
id	Int	11	Primary key of admin table
username	Varchar	50	To store the username of admin
password	Varchar	50	Store password of the admin

Table No 02 : category**Description : store the category of products****Primary key :id**

Field Name	Type	Size	Description
id	Int	11	Primary key of category table
categoryName	Varchar	30	To store category name
categoryDescription	longtext		To store description of category

Table No 03 : Subcategory**Description : To store the subcategory of products.****Primary key :id****Foreign Key:categoryid from category table.**

Field Name	Type	Size	Description
id	Int	11	Primary key of subcategory
categoryid	Int	11	Foreign key of
subcategory	Varchar	30	To store subcategory names

Table No 04 : products**Primary key :product_id****Description : To store the product details.**

Field Name	Type	Size	Description
id	Int	11	Primary key of Products table
category	Int	11	Category of product table
subcategory	Int	11	Subcategory of product
productName	Varchar	50	Title of product
productPrice	Varchar	11	Price of product
productQuantity	Varchar	11	Number of product quantity
productImage1	Varchar	150	First Images of product
productImage2	Varchar	150	Second Images of product
productImage3	Varchar	150	Third image of product
shippingCharge	Int	11	Shipping charge of product
productAvailiability	Varchar	15	Information about product stock or out of stock

Table No 05 : users**Primary key :id****Description: To store the user details.**

Field Name	Type	Size	Description
id	Int	11	Primary key of User table
name	Varchar	50	Name of user
email	Varchar	60	Email adress of user
contactno	Int	12	Contact number of user
u_type	Varchar	10	Login user type
password	Varchar	25	Password of user login
shippingAddress	longtext		shippingAddress of user for product deluivery
shippingState	Varchar	25	State of product delivery
shippingCity	Varchar	25	City of product delivery
billingAdress	longtext		Billing address of product delivery
billingState	Varchar	25	Billing state of product delivery
billingCity	Varchar	25	Billing city of product delivery
billingPincode	Int	11	Pincode of product delivery

Table No 06 : orders**Primary key :id****Foreign key: userID ,productId from user and product tables.****Description : To store the order details.**

FieldName	Type	Size	Description
id	Int	11	Primary key of Orders table
userID	Int	11	Foreign key of user table
ProductId	Varchar	50	Foreign key of products table
quantity	Int	11	Quantity of ordered product
orderDate	Date		Date of ordering
orderStatus	Varchar	50	Status of ordered product
total	Int	20	Total amount of order

Table No 07 : wishlist**Primary key :id****Foreign key: productId,userID from user and products tables.****Description : To store the wished items.**

Field Name	Type	Size	Description
id	Int	11	Primary key of wishlist table
productId	Int	11	Foreign key of products table
userID	Int	11	Foreign key of user table

Table No 08 : productreviews**Primary key :id****Foreign key:productId from product table.****Description : To store the product reviews of users.**

Field Name	Type	Size	Description
id	Int	11	Primary key of productreviews table
productId	Int	11	Foreign key of products table
quantity	Int	11	Number of ordered quantity
price	Int	11	Price of ordered product
value	Int	11	Number of star giving the product
name	Varchar	50	Name of review added user
summary	Varchar	100	Summary of product
review	longtext		Review about the product

Table No 09: ordertrackhistory**Primary key :id****Foreign key:orderId from order table.****Description : To store the order track history.**

Field Name	Type	Size	Description
id	Int	11	Primary key of Ordertrackinghistory table
orderId	Int	11	Foreign key of orders table
status	Varchar	50	Status of user ordered product
remarkt	Varchar	100	Staus description

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and as well. Checking that what has been specified is what the user requested is the process of validation. The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running in order to offer data like execution traces, timing profiles, and test coverage details. Testing is a collection of activities that can be planned ahead of time and carried out in a methodical manner. Testing starts with individual modules and progresses all the way to system integration for computer-based systems. There are many rules that can be used as testing objectives, and testing is necessary for the system testing objectives to be successful. As follows: A programme is tested by being run with the goal of identifying any errors.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be

anticipated and error handling paths set to reroute or cleanly terminate processing when error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. Once these mistakes are fixed, new ones come into being, and the cycle repeats itself indefinitely. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. In addition, variations in programme structures were eliminated, and a special programme structure developed.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

5.2.4 Output Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points

- Output Screen Designs,
- Input Screen Designs

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again covered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the development of the product. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarizes what was discovered, and this data can be compared to results from earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Test cases for different scenarios are carefully built for automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are simply impossible for a human to produce. Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This procedure can be made simpler with automation testing. Additionally, it can examine a greater test coverage than perhaps people can.

- Simplifies testing - Most SaaS and tech organizations regularly test their products as part of daily operations. The key is to keep things as basic as you can. Automation has a lot of advantages. The test scripts can be reused when automating test tools.
- Speeds up the testing process - Humans cannot keep up with automated technology and machines. This is why we employ them, along with increased accuracy. Your software development cycles are subsequently shortened by this.

5.2.6 Selenium Testing

An open-source programme called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in a number of different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. Web application testing for cross-browser compatibility is automated using the Selenium testing tool.

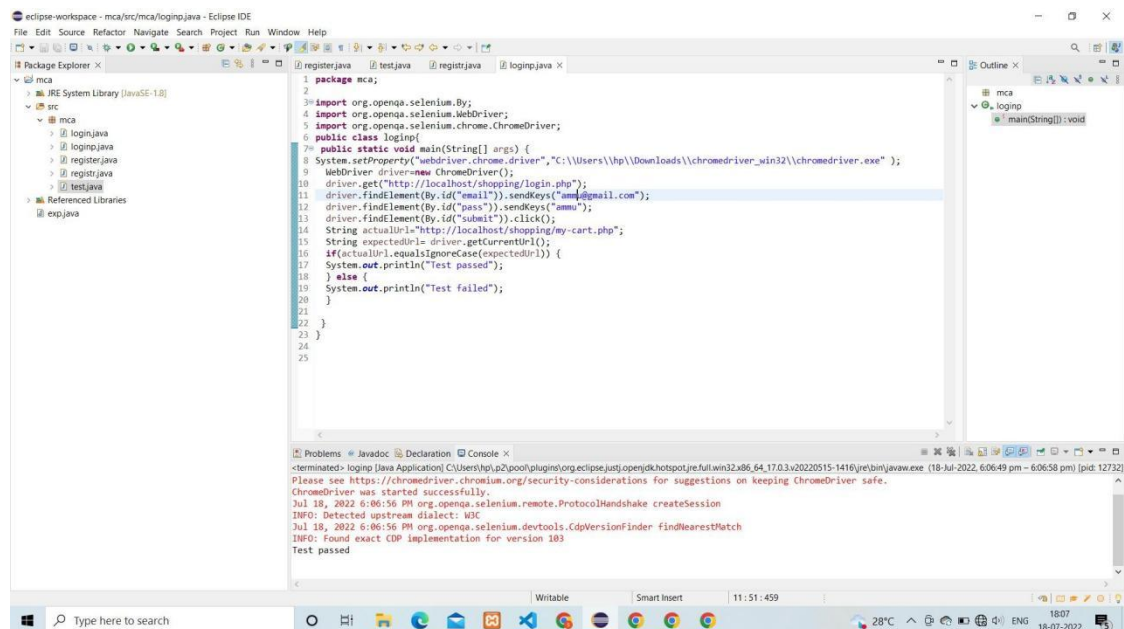
Test cases for a Login Page

Test Case ID: Fun_1			Test Designed By: Hima M S		
Test Priority (Low/Medium/High): High			Test Designed Date: 18-07-2022		
Module Name: Login Screen			Test Executed By: Ms Rini Kurian		
Test Title: Verify login with validusername and password			Test Execution Date: 18-05-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	ExpectedResult	Actual Result	Status(Pass/Fail)
1	Navigation to LoginPage		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid User Name	User Name: ammu@gmail.com	User shouldbe able to Login	User Logged in andnavigated to the dashboard with records	Pass
3	Provide Valid Password	Password: ammu			
4	Click on Sign In button				
5	Provide Invalid user name or Password	Email Id: amu@gmail.com Password: ammu	User should notbe able to Login	Message for enter valid username or password displayed	Pass
6	Provide Null Username Id orPassword	Email Id: null Password: null			
7	Click on Sign In button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

LOGIN

login package mca;

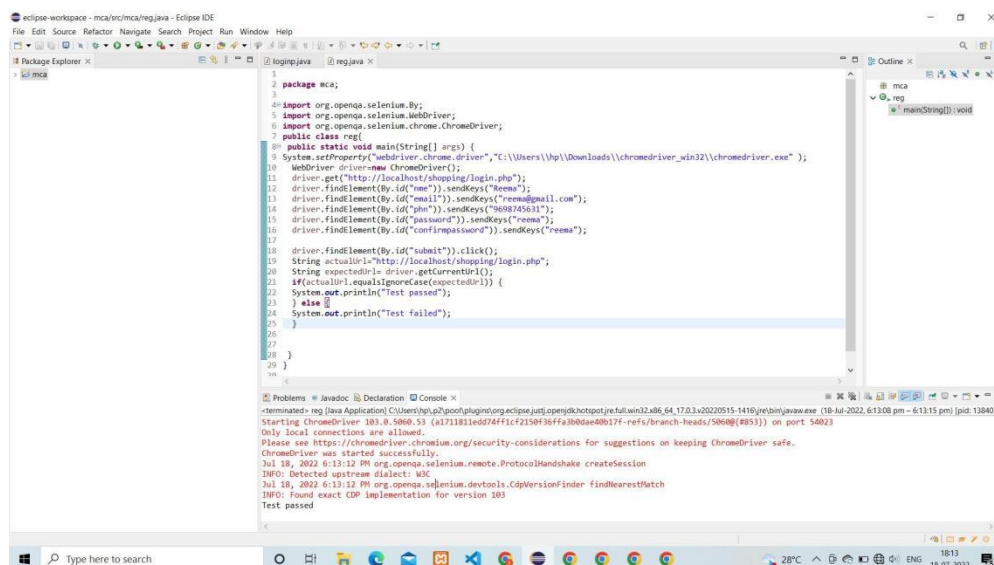
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;public
class loginp {
public static void main(String[] args)
{ System.setProperty("webdriver.chrome.driver","C:\\Users\\hp\\Downloads\\chromedriver_
wi32\\chromedriver.exe"); WebDriver
driver=new ChromeDriver();
driver.get("http://localhost/shopping/login.php");
driver.findElement(By.id("email")).sendKeys("ammu@gmail.com");
driver.findElement(By.id("pass")).sendKeys("ammu"); driver.findElement(By.id("submit")).click();
String actualUrl="http://localhost/shopping/my-cart.php";String
expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl))
{ System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
}
```



Test cases for customers Registration

Project Name: Online Horticulture Shop					
Updation Test Case					
Test Case ID: registration			Test Designed By: Hima m s		
Test Priority (Low/Medium/High): High			Test Designed Date: 18-07-2022		
Module Name: Register Screen			Test Executed By:Ms.Rini Kuriann		
Test Title: User Registration Details			Test Execution Date: 18-07-2022		
Description: Register to system and Registration is completed then login , if some error occurs, test will fail					
Pre-Condition: User has valid user name and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Register Page		Register Page Should be displayed	Registration page displayed	Pass
2	Provide Valid Registration details	First Name: Reema Email:reema@gmail.com Mobile: 9967341098 Password: reema User Type: customer	User shouldbe able to Register	User registration Completed after go to the login page	Pass
3	Provide valid password	Password: reema			
4	Click on gister button				
5	Provide invalid username or password	User Name:reema@gmail.co Password: abcd	User should not able to login	User will not redirected to next page.	Pass
6	Provide Null username or Password	User Name: null Password: null			
7	Click on Register button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

```
public class reg {  
    public static void main(String[] args) {  
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\hp\\Downloads\\chromedriver_win32\\c h  
            medriver.exe" );  
        WebDriver driver=new ChromeDriver();  
        driver.get("http://localhost/shopping/login.php");  
        driver.findElement(By.id("nme")).sendKeys("Reema");  
        driver.findElement(By.id("email")).sendKeys("reema@gmail.com");  
        driver.findElement(By.id("phn")).sendKeys("9698745631");  
        driver.findElement(By.id("password")).sendKeys("reema");  
        driver.findElement(By.id("confirmpassword")).sendKeys("reema");  
  
        driver.findElement(By.id("submit")).click();  
        String actualUrl="http://localhost/shopping/login.php";  
        String expectedUrl= driver.getCurrentUrl();  
        if(actualUrl.equalsIgnoreCase(expectedUrl))  
        { System.out.println("Test passed");  
        } else {  
            System.out.println("Test failed");  
        }  
    }  
}
```



CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining the users' trust that the new system will function, be efficient, and be accurate can be the most important step in creating a successful new system. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.. At this point, the user department is responsible for the majority of the workload, the most disruption, and the most influence on the current system. If the implementation is not well thought out or managed, confusion and disorder may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organisational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards, can it be put into use. The system personnel assess the system's viability. The effort necessary for system analysis and design to implement the three key components of education and training, system testing, and changeover will increase in proportion to how complicated the system being implemented is. The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about software but we have to ensure that the resistance does not build, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, reply to error alerts, query the database, and call up routines that will generate reports and carry out other necessary tasks through user training.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the data entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

6.2.4 Hosting

app.infinityfree.net is a free website hosting platform that provides an array of valuable features, including a website builder, control panel and file manager.

Hosting Procedures

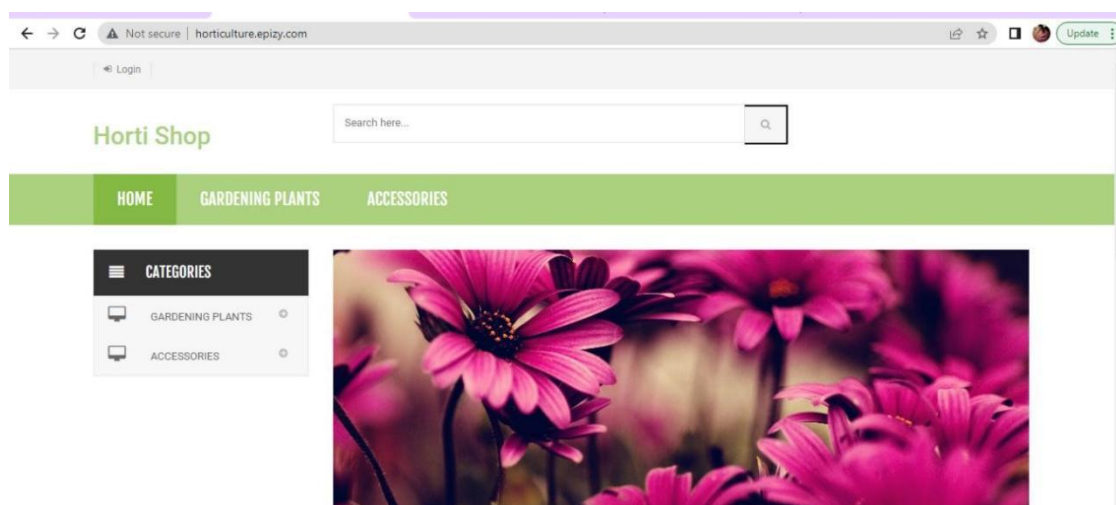
1. Create an Account or Log In into your account.
2. Create a Hosting Account.
3. Create a Custom Domain or a Subdomain provided by InfinityFree.
4. Manage your account.
5. Upload your Files.
6. Creating Your Database.
7. Changing your PHP connection file configuration.

The screenshot displays the InfinityFree Control Panel. At the top, there are two main buttons: 'Control Panel' (green) and 'File Manager' (orange). Below these, the interface is divided into several sections:

- Account Details:** A table showing account information.

Username	epiz_32216636
Password	***** Show/Hide
Status	Active
Label	horticulture-online shopping
Main Domain	bwm16eqg.epizy.com
Website IP	185.27.134.149
Hosting Volume	vol1_1
Created on	2022-07-21
- FTP Details:** A table showing FTP account information.

FTP Username	epiz_32216636
FTP Password	***** Show/Hide
FTP Hostname	ftpupload.net
FTP Port (optional)	21
- Account Settings:** A sidebar with buttons for 'Edit Account', 'Deactivation History' (with a red indicator), and 'Redirects'.
- MySQL Details:** A section at the bottom left.
- Domains and Subdomains:** A section at the bottom right.



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

To overcome all the limitations of the existing system, 'Online Horticulture Shop' was developed. it is the best possible solution for the existing system. The administrator only has the right to access all the data in the system. All the requirements are built into the system. It produces data output in the required format. Our project is only a humble venture to satisfy the needs to manage their project work. Several friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements. The objective of software planning is to provide a framework that enables the admin to make responsible estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

7.2 FUTURE SCOPE

In future we can expect the modified version of 'Online Horticulture Shop'. The system is very flexible for further up gradation with additional requirement of the women, the PHP and MYSQL make this modification very easily it is also possible to involve more functions into the system.

- Add more categories of products .
- Data security can be enhanced.
- Add more delivery boys and enhance the system.
- Salary distribution of delivery boy and admin.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- <https://www.tutorialspoint.com/mysql/>
- <https://www.JSP.net/>
- <https://www.xampp.com/en/>
- <https://www.tutorialspoint.com/java/>
- <https://www.javatpoint.com/javatutorial/>

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.php

```
<?php
session_start();
error_reporting
(0);
include('includes/config.php');
// Code user Registration
if(isset($_POST['submit']))
{
$name=$_POST['fullname'];
$email=$_POST['emailid'];
$contactno=$_POST['contactno'];
$password=(($_POST['password']));
$u_type=$_POST['u_type'];
$query=mysqli_query($con,"insert into
users(name,email,contactno,password,u_type)
values('$name','$email','$contactno','$password','$u_type')");
if($query)
{
echo "<script>alert('You are successfully register');</script>";
}
else{
echo "<script>alert('Not register something went worng');</script>";
}
}

// Code for User login
if(isset($_POST['login']))
```

```
{
$email=$_POST['email'];

$password=( $_POST['password']);

$query=mysqli_query($con,"SELECT * FROM users WHERE email='$email' and
password='$password'");

$num=mysqli_fetch_array($query);

if($num>0)

{

//$extra="my-cart.php";

$_SESSION['login']=$_POST['email'];

$_SESSION['id']=$num['id'];

$_SESSION['username']=$num['name'];

if($num['u_type']=='stock')

{

header("Location:stock/dash.php");

}

else if($num['u_type']=='dboy')

{

header('location:dboy/today-order.php');

}

else if($num['u_type']=='customer')

{

//$extra="my-cart.php";

header("Location:my-
cart.php");

}

}

}
```

Addproduct.php

```
<?php
session_
start();
include('include/config.php');
if(strlen($_SESSION['alogin'])==0)
{
header('location:index.php');
}
else{

if(isset($_POST['submit']))
{
$category=$_POST['category'];
$subcat=$_POST['subcategory'];
$productname=$_POST['productName'];
$productcompany=$_POST['productCompany'];
$productprice=$_POST['productprice'];
$productpricebd=$_POST['productpricebd'];
$productdescription=$_POST['productDescription'];
$productcharge=$_POST['productShippingcharge'];
$productavailability=$_POST['productAvailability'];
$productimage1=$_FILES["productimage1"]["name"];
$productimage2=$_FILES["productimage2"]["name"];
$productimage3=$_FILES["productimage3"]["name"];
//for getting product id
$query=mysqli_query($con,"select max(id) as pid from products");
$result=mysqli_fetch_array($query);
$productid=$result['pid']+1;
$dir="productimages/$productid";
if(!is_dir($dir)){ mkdir("productimages/".$
$productid);
}

move_uploaded_file($_FILES["productimage1"]["tmp_name"],"productimages/$productid/".$_FIL
ES["productimage1"]["name"]);
move_uploaded_file($_FILES["productimage2"]["tmp_name"],"productimages/$productid/".$_FIL
ES["productimage2"]["name"]);
move_uploaded_file($_FILES["productimage3"]["tmp_name"],"productimages/$productid/".$_FIL
ES["productimage3"]["name"]);
$sql=mysqli_query($con,"insert
hi
ppingCharge,productAvailability,productImage1,productImage2,productImage3,productPriceBefor
e Discount)
into
products(category,subCategory,productName,productCompany,productPrice,productDescription,s
```

```
tpricebd')));
$_SESSION['msg']="Product Inserted Successfully !!";

}

?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Admin| Insert Product</title>
<link type="text/css" href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link type="text/css" href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">
<link type="text/css" href="css/theme.css" rel="stylesheet">
<link type="text/css" href="images/icons/css/font-awesome.css" rel="stylesheet">
<link type="text/css"
href='http://fonts.googleapis.com/css?family=Open+Sans:400italic,600italic,400,600'
rel='stylesheet'>
<script src="http://js.nicedit.com/nicEdit-latest.js" type="text/javascript"></script>
<script type="text/javascript">bkLib.onDomLoaded(nicEditors.allTextAreas);</script>

<script>
function getSubcat(val) {
$.ajax({
type: "POST",
url:
"get_subcat.php",
data:'cat_id='+val,
success:
function(data){
$("#subcategory").html(data);
}
});
}
function selectCountry(val) {
$("#search-box").val(val);
$("#sugesstion-box").hide();
}
</script>

</head>
<body>
<?php include('include/header.php');?>

<div class="wrapper">
<div class="container">
<div class="row">
<?php include('include/sidebar.php');?>
```



```
<div class="module">
<div class="module-head">
<h3>Insert Product</h3>
</div>
<div class="module-body">

<?php if(isset($_POST['submit']))
{?>
<div class="alert alert-success">
<button type="button" class="close" data-dismiss="alert">×</button>
<strong>Well done!</strong> <?php echo htmlentities($_SESSION['msg']);?><?php echo
htmlentities($_SESSION['msg']="");?>
</div>
<?php } ?>

<?php if(isset($_GET['del']))
{?>
<div class="alert alert-error">
<button type="button" class="close" data-dismiss="alert">×</button>
<strong>Oh snap!</strong> <?php echo
htmlentities($_SESSION['delmsg']);?><?php echo
htmlentities($_SESSION['delmsg']="");?>
</div>
<?php } ?>

<br />

<form class="form-horizontal row-fluid" name="insertproduct"
method="post" enctype="multipart/form-data">

<div class="control-group">
<label class="control-label" for="basicinput">Category</label>
<div class="controls">
<select name="category" class="span8 tip" onChange="getSubcat(this.value);" required>
<option value="">Select Category</option>
<?php $query=mysqli_query($con,"select * from
category");while($row=mysqli_fetch_array($query))
{?>

<option value="<?php echo $row['id'];?>"><?php echo $row['categoryName'];?></option>
<?php } ?>
</select>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Sub Category</label>
<div class="controls">
```

```
</select>
</div>
</div>
<div class="control-group">
<label class="control-label" for="basicinput">Product Name</label>
<div class="controls">
<input type="text" name="productName" placeholder="Enter Product Name" class="span8
tip"required>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Product Company</label>
<div class="controls">
<input type="text" name="productCompany" placeholder="Enter Product Comapny
Name"class="span8 tip" required>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Product Price Before Discount</label>
<div class="controls">
<input type="text" name="productpricebd" placeholder="Enter Product Price" class="span8
tip"required>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Product Price After Discount(Selling Price)</label>
<div class="controls">
<input type="text" name="productprice" placeholder="Enter Product Price" class="span8
tip"required>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Product Description</label>
<div class="controls">
<textarea name="productDescription" placeholder="Enter Product Description" rows="6"
class="span8 tip">
</textarea>
</div>
</div>

<div class="control-group">
<label class="control-label" for="basicinput">Product Shipping Charge</label>
<div class="controls">
<input type="text" name="productShippingcharge" placeholder="Enter Product Shipping
Charge"class="span8 tip" required>
</div>
```

```
</div>

<div class="control-group">
  <label class="control-label" for="basicinput">Product Availability</label>
  <div class="controls">
    <select name="productAvailability" id="productAvailability" class="span8 tip" required>
      <option value="">Select</option>
      <option value="In Stock">In Stock</option>
      <option value="Out of Stock">Out of Stock</option>
    </select>
  </div>
</div>

<div class="control-group">
  <label class="control-label" for="basicinput">Product Image1</label>
  <div class="controls">
    <input type="file" name="productimage1" id="productimage1" value="" class="span8 tip"
    required>
  </div>
</div>

<div class="control-group">
  <label class="control-label" for="basicinput">Product Image2</label>
  <div class="controls">
    <input type="file" name="productimage2" class="span8 tip" required>
  </div>
</div>

<div class="control-group">
  <label class="control-label" for="basicinput">Product Image3</label>
  <div class="controls">
    <input type="file" name="productimage3" class="span8 tip">
  </div>
</div>

<div class="control-group">
  <div class="controls">
    <button type="submit" name="submit" class="btn">Insert</button>
  </div>
</div>
</form>
</div>
</div>
```

```

</div><!--/.content-->
</div><!--/.span9-->
</div>
</div><!--/.container-->
</div><!--/.wrapper-->

<?php include('include/footer.php');?>

<script src="scripts/jquery-1.9.1.min.js" type="text/javascript"></script>
<script src="scripts/jquery-ui-1.10.1.custom.min.js" type="text/javascript"></script>
<script src="bootstrap/js/bootstrap.min.js" type="text/javascript"></script>
<script src="scripts/flot/jquery.flot.js" type="text/javascript"></script>
<script src="scripts/datatables/jquery.dataTables.js"></script>
<script>
$(document).ready(function() {
$('.datatable-1').dataTable();
$('.dataTables_paginate').addClass("btn-group datatable-pagination");
$('.dataTables_paginate > a').wrapInner('<span />');
$('.dataTables_paginate > a:first-child').append('<i class="icon-chevron-left shaded"></i>');
$('.dataTables_paginate > a:last-child').append('<i class="icon-chevron-right shaded"></i>');
} );
</script>
</body>
<?php } ?>

<div class="col-md-7">
<div class="card">
<div class="card-header card-header-primary">
<h5 class="title">Add Product</h5>
</div>
<div class="card-body">
<div class="row">
<div class="col-md-12">
<div class="form-group">
<label>Product Name</label>
<b><input type="hidden" name="product_id" class="form-control" value="<?php
echoisset($_GET['id']) ? $_GET['id'] : " ?>"></b>
<input type="text" id="product_name" required name="product_name" class="form-control"
value="<?php echo isset($meta['product_title']) ? $meta['product_title'] : " ?>">
</div>
</div>
<div class="col-md-4">
<div class="">
" alt="" class="img-field" width="75" height="100">

<label for="">Product Image</label>
<input type="file" name="picture" <?php echo !isset($meta['product_image']) ? 'required' : " ?>

```

```

<class="btn btn-fill" id="picture" onchange="displayImg(this,$(this))">
</div>
</div>
<div class="col-md-12">
<div class="form-group">
<label>Description</label>
<textarea rows="4" cols="80" id="details" required name="details" class="form-
control"><?php echo isset($meta['product_desc']) ? $meta['product_desc'] : " ?"></textarea>
</div>
</div>
</div>
<div class="card">
<div class="card-header card-header-primary">
<h5 class="title">Categories</h5>
</div>
<div class="card-body">

<div class="row">

<div class="col-md-12">
<div class="form-group">
<label>Product Category</label>
<select name="category_id" id="category_id" class="default-browser custom-select select2">
<option value=""></option>
<?php
$cat = mysqli_query($con,"select *
fromcategories");while($row =
mysqli_fetch_assoc($cat)):
?>
<option value="<?php echo $row['cat_id'] ?>" <?php echo isset($meta['product_cat']) &&
$meta['product_cat'] == $row['cat_id'] ? 'selected' : " ?">><?php echo $row['cat_title'] ?></option>
<?php endwhile; ?>
</select>
</div>
</div>
<div class="col-md-12">
<div class="form-group">
<label for="">Product Brand</label>
<select name="brand_id" id="brand_id" class="default-browser custom-select select2">
<option value=""></option>
<?php
$cat = mysqli_query($con,"select * from
brands");while($row =
mysqli_fetch_assoc($cat)):
?>
<option value="<?php echo $row['brand_id'] ?>" <?php echo isset($meta['product_brand'])

```

```

$row['brand_title'] ?></option>
<?php endwhile; ?>
</select>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="card-footer">
<button type="submit" id="btn_save" name="btn_save" required class="btn btn-fill btn-
primary">Save Product</button>
</div>
</div>
</div>

</div>
</form>

</div>
</div>
<script>
$(.select2').select2({ placeholder:
"Pleas
e
select
Here",
width:'
100%'
})

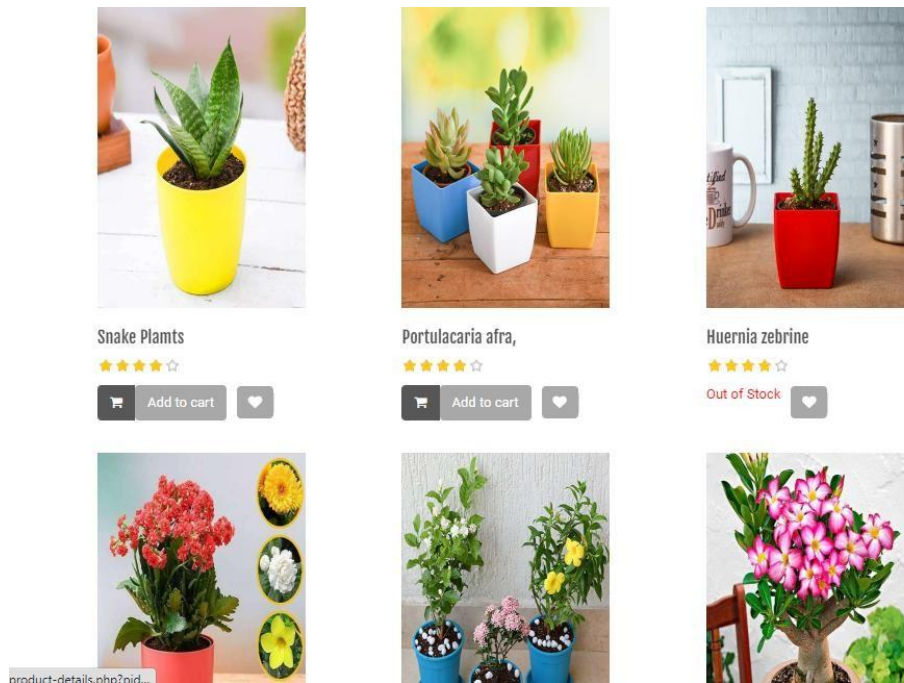
$('#manage-
prod').submit(function(e){ e.preventDefault()start_load()
$.ajax({ u
rl:'save_
prod.php',
data: new
FormData($(this)[0]),
cache: false,
contentType
e: false,
processDat
a: false,
method:
'POST',
type:
'POST',
error:err=>console.
log(err),
success:function(re

```

```
sp){ if(resp  
== 1){  
alert("Data successfully saved.");  
location.replace('index.php?page=productlist')  
}  
{  
}  
}  
}  
</script>
```

9.2 SCREENSHOT

CUSTOMER :HOME



DELIVERY BOY:STATUS UPDATE

Update Order !

order Id: 6

Status: Delivered ▼

Remark:

update

Close this Window

ADMIN:INSERT PRODUCT

Order Management
Manage users
Create Category
Sub Category
Insert Product
Manage Products
User Login Log
Logout

Insert Product

Category
Select Category

Sub Category

Product Name
Enter Product Name

Product Company
Enter Product Comapny Name

Product Price Before Discount
Enter Product Price

Product Price After Discount(Selling Price)
Enter Product Price

Product Description

B I U
Font Size...
Font Familv...
Font Format

STOCK MANAGEMENT:DASHBORD

Dashboard
Feed Back
Report

Total Product
21

Total Orders
6

Low Stock
1

18
Monday 18, 2022
Add DELIVERY BOY

Calendar
July 2022

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23

PAPER NAME

ONLINE HORTICULTURE SHOP .docx

AUTHOR

Hima M S

WORD COUNT

7883 Words

CHARACTER COUNT

41762 Characters

PAGE COUNT

52 Pages

FILE SIZE

2.0MB

SUBMISSION DATE

Jul 22, 2022 8:38 AM GMT+5:30

REPORT DATE

Jul 22, 2022 8:39 AM GMT+5:30

● 24% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 24% Internet database
- 0% Publications database
- Crossref database

● Excluded from Similarity Report

- Small Matches (Less than 8 words)

● 24% Overall Similarity

Top sources found in the following databases:

- 24% Internet database
- 0% Publications database
- Crossref database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	issuu.com Internet	7%
2	tutorialspoint.com Internet	2%
3	pdfcoffee.com Internet	2%
4	ia801708.us.archive.org Internet	2%
5	docshare.tips Internet	1%
6	geeksforgeeks.org Internet	1%
7	slideshare.net Internet	<1%
8	coursehero.com Internet	<1%
9	documents.mx Internet	<1%

10	dev.mysql.com Internet	<1%
11	studymode.com Internet	<1%
12	scribd.com Internet	<1%
13	cloud-hosting.io Internet	<1%
14	seminarprojects.com Internet	<1%
15	dspace.cusat.ac.in Internet	<1%
16	seminarprojects.com Internet	<1%
17	irjmets.com Internet	<1%
18	kupdf.net Internet	<1%
19	elegantmicroweb.com Internet	<1%
20	vsbec.com Internet	<1%
21	cs.worcester.edu Internet	<1%

22	thesoftwareguild.com	Internet	<1%
23	idr.mnit.ac.in	Internet	<1%
24	docplayer.net	Internet	<1%
25	justanswer.com	Internet	<1%
26	dspace.daffodilvarsity.edu.bd:8080	Internet	<1%
27	eml.ou.edu	Internet	<1%
28	hurson.weebly.com	Internet	<1%
29	inba.info	Internet	<1%
30	wikieducator.org	Internet	<1%
31	classle.net	Internet	<1%
32	engineeringminiprojects.com	Internet	<1%
33	ukessays.com	Internet	<1%