ONLINE PLANTS NURSERY

Project Report Submitted By

MANEEKSHA C ASHOK

Reg.No: AJC17MCA-I035

In Partial fulfillment for the Award of the Degree Of

INTEGRATED MASTER OF COMPUTER APPLICATIONS (INMCA)
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

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

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**ONLINE PLANTS NURSERY”** is the bonafide work of **MANEEKSHA C ASHOK (Reg.No: AJC17MCA-I035)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2017-2022.

Ms. Shelly Shiju George
Internal Guide

Ms. Meera Rose Mathew
Coordinator

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

External Examiner

DECLARATION

I hereby declare that the project report **"ONLINE PLANTS NURSERY"** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021-2022.

Date:	**MANEEKSHA C ASHOK**
KANJIRAPPALLY	**Reg. No: AJC17MCA-I035**

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. Lillykuttty 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 coordinator **Ms. Meera Rose Mathew** for her 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. Shelly Shiju George** 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.

MANEEKSHA C ASHOK

ABSTRACT

The Online Plants Nursery system is an online platform for buying plants & gardening items for making our surroundings eco-friendly, which is most important segment in setting the complete ambience of our home. The system consists of all gardening products such as a wide variety of plants, fertilizers, containers, gardening tools etc. The user can view the specification of each product. The registered user can buy product by adding it into the cart along with the quantity of the product & payment can be done through different methods like online transactions and cash on delivery.

CONTENT

List of Abbreviation

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 Plants Nursery" project is an online platform for buying plants & gardening items for making our surroundings eco-friendly, which is most important segment in setting the complete ambience of our home. The system consists of all gardening products such as a wide variety of plants, fertilizers, containers, gardening tools etc. The user can view the specification of each product. The registered user can buy product by adding it into the cart along with the quantity of the product & payment can be done through different methods like online transaction and cash on delivery.

1.2 PROJECT SPECIFICATION

The proposed system contains the following features: User can view the specification of each product. The registered user can buy product by adding it into the cart along with the quantity of the product & payment can be done through different methods like online transaction and cash on delivery.

The system is divided into mainly two modules they are:

1. **Admin Module**

   Managing all the stocks. Admin can view the product details and update the details of the product. Adding new products and new categories of products and services. Monitoring the feedback from each and every one and rectifying their issues over the site and other things. Admin can view the summary of the user details.

2. **End-Users Module**

   The user module mainly divided into two:

   a. **Non registered user:**

   They are only allowed to surf through the system, they are limited to access for the site full potentials. Moreover, additional functionalities such as Purchasing products can only be available when the user registered to the website.

   b. **Registered user:**

   They are actual users of the system and they are provided with the functionalities such as: These users can store their profile information on the site and can change or update their details any time. They can search for the desired product and buy it from the site by adding it in to their own cart. However, to become a registered user one must register through the registration link or the signup section and the sign in to use these functionalities. Moreover, additional functionalities such as buying specific products and have the right to send feedbacks.

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 decisional 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

A nursery is a place where plants are propagated and grown to a desired age. Mostly the plants concerned are for gardening, forestry or conservation biology, rather than agriculture. They include retail nurseries, which sell to the general public, wholesale nurseries, which sell only to businesses such as other nurseries and to commercial gardeners, and private nurseries, which supply the needs of institutions or private estates. Some will also work in plant breeding. Some produce bulk stock, whether seedlings or grafted, of particular varieties for purposes such as fruit trees for orchards, or timber trees for forestry.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient
- Human effort is needed.

2.4 PROPOSED SYSTEM

In Online plants nursery the client must approach the web. Web-based shopping is normally accessible 24 hours per day. The proposed system contains the following features: User can view the specification of each product. The registered user can buy product by adding it into the cart along with the quantity of the product & payment can be done through different methods like online transaction and cash on delivery.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Finding a plant online is considerably less demanding than searching for it in the neighborhood store.
- It is simpler to discover uncommon plants shopping on the web, as it is extremely valuable in purchasing uncommon plants
- Plants are frequently more modest. Plants are regularly less expensive online than they are in stores.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

The cost of project was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open-source software.

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

3.1.3 Behavioral Feasibility

The proposed system includes the following 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.

Online Plant Nursery, GUI is simple so that users can easily use it. **Online Plant Nursery** is simple enough so that no training is needed.

3.2 SYSTEM SPECIFICATION

3.2.2 Hardware Specification

RAM 512MB and Above

HDD 40 GB Hard Disk Space and Above

Processor - Intel core i3

3.2.3 Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC – Windows 7 and above

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.2 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Hypertext Preprocessor, a recursive acronym code is interpreted by a web server with a PHP processor module which generates the resulting web page commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.1.2 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

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

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986.

- **MySQL software is Open Source**

  Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **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.**

  The MySQL Database Software is a client/server system that consists of a multi- threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

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

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation forth modeling of real-world objects and systems.

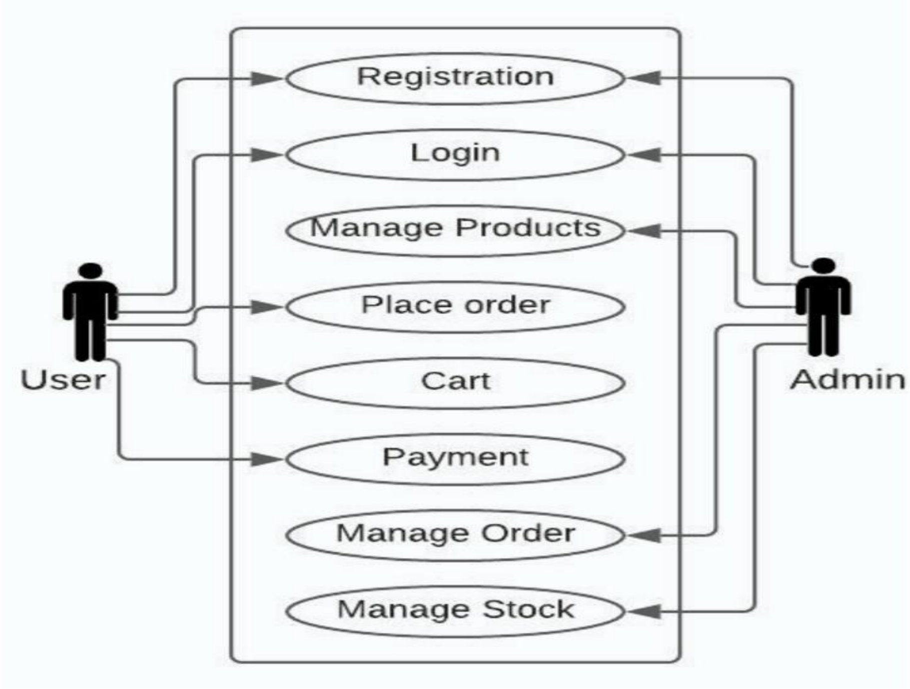
System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains fourcomponents.

- 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.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1: Use case diagram for **online plant nursery**



4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

I. **Actors –** An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

II. **Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically, each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

III. **Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

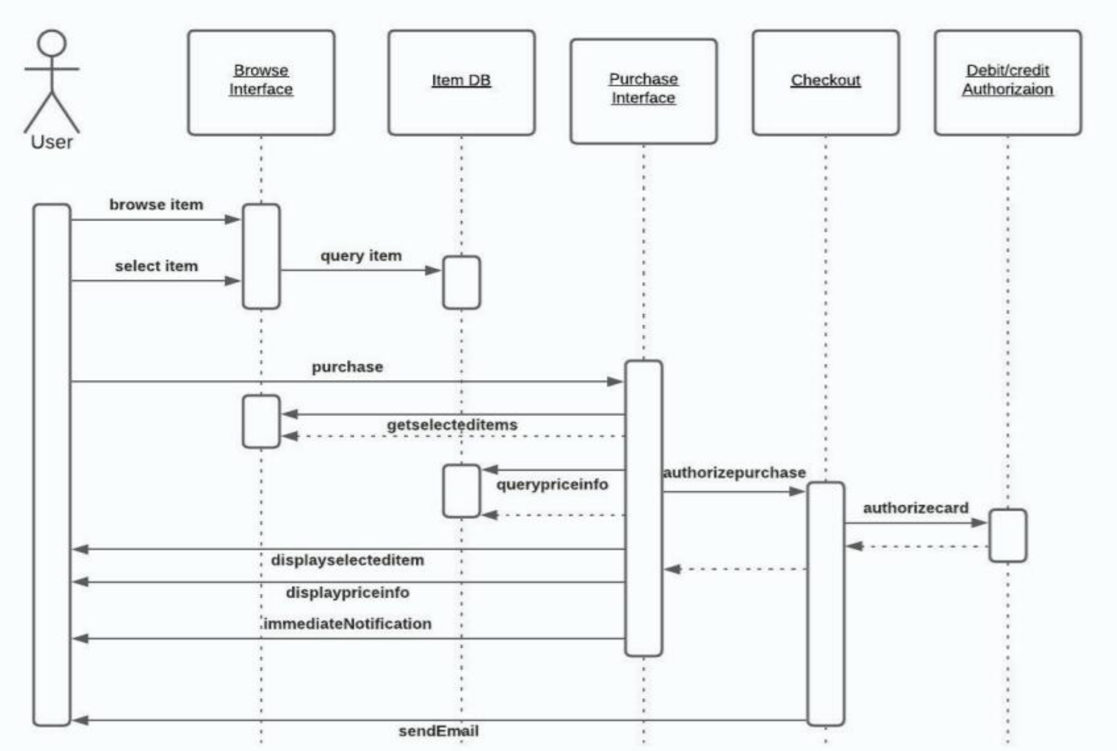
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

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 sophisticate 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.

Fig 2: Sequence diagram of user in **online plant nursery**



4.2.3 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.

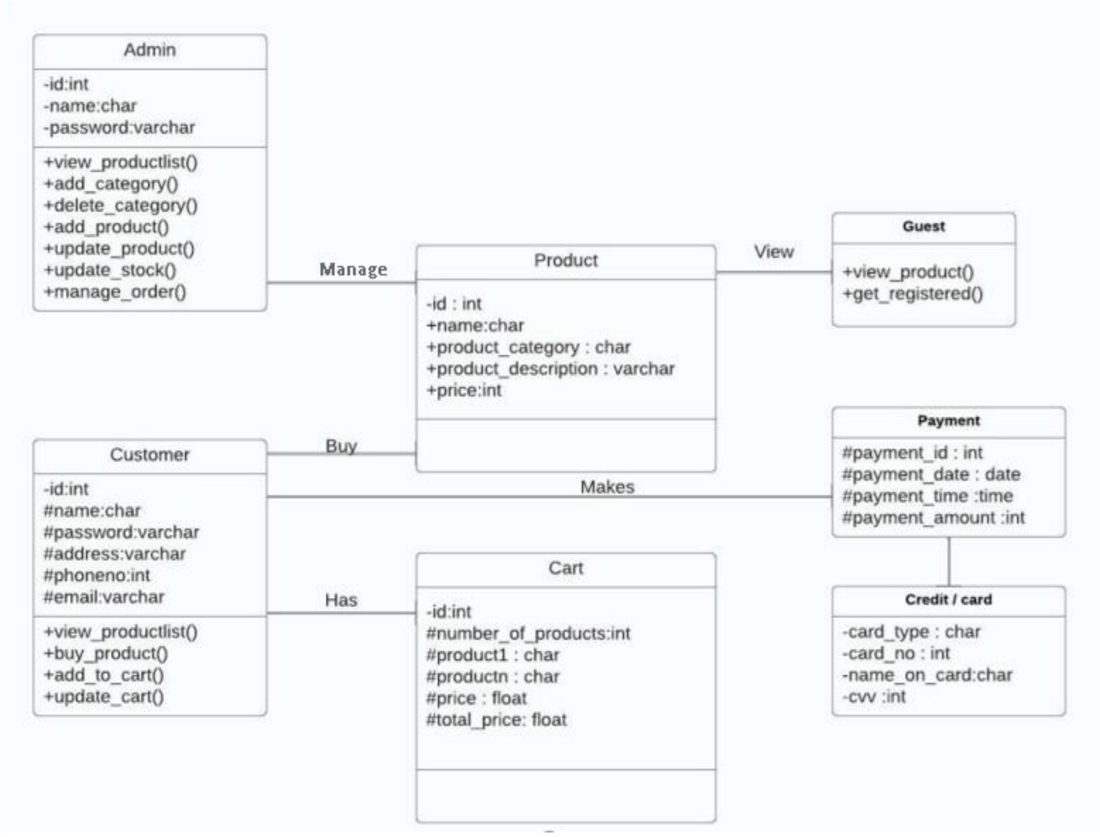
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as −

- Analysis and
- design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Fig 3: Class diagram for **online plant nursery**



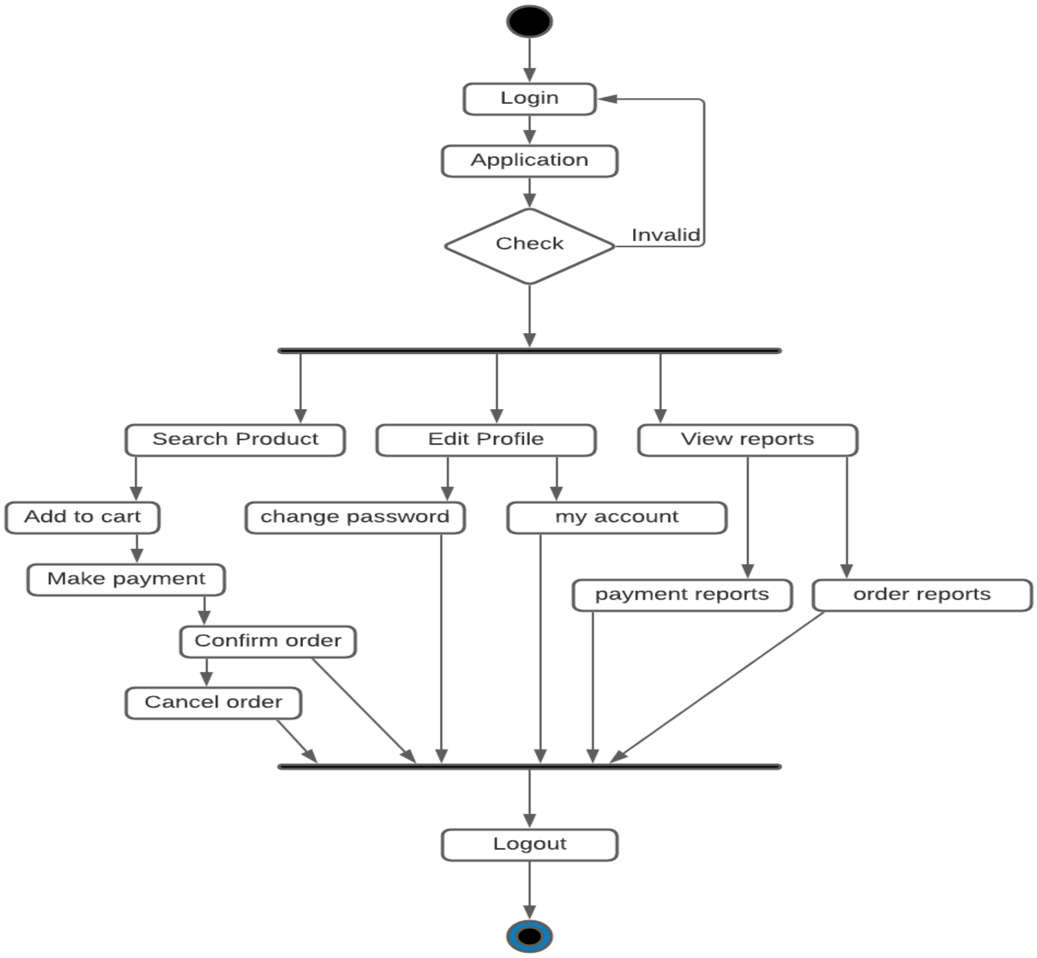
4.2.4 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

Fig 4: Activity diagram for **online plant nursery**



4.2.5 COMPONENT DIAGRAM

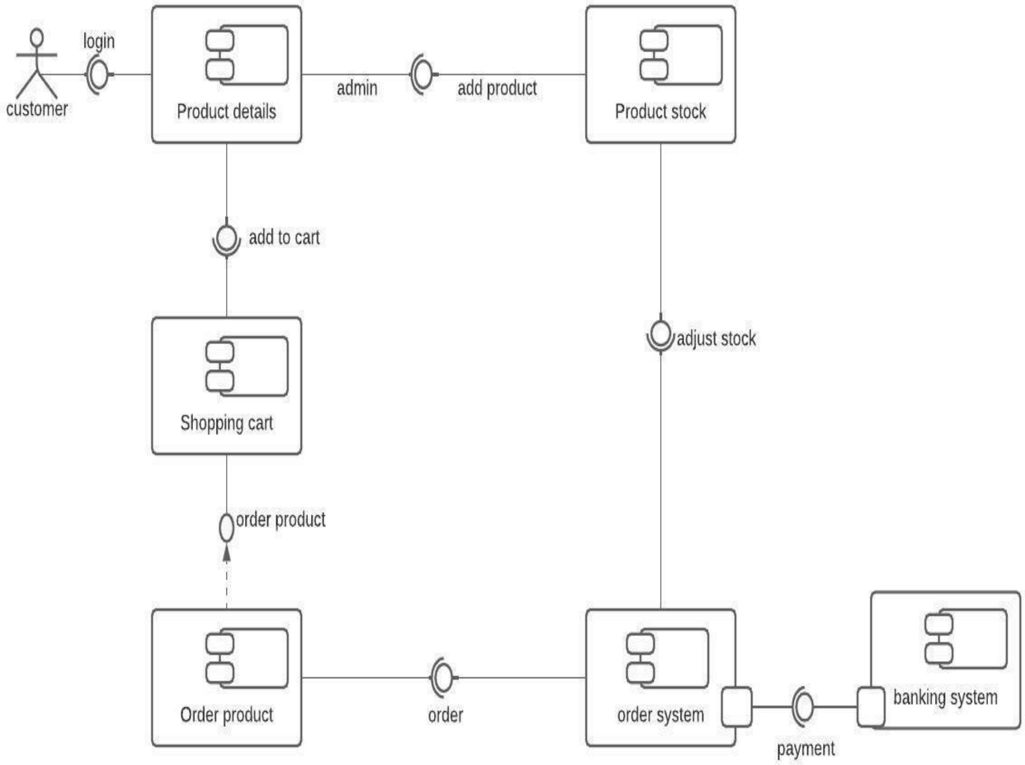
Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Component diagrams can also be described as a static implementation view of a system.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executables by using forward and reverse engineering.
- Describe the organization and relationships of the components.

Fig 5: Component diagram for **online plant nursery**



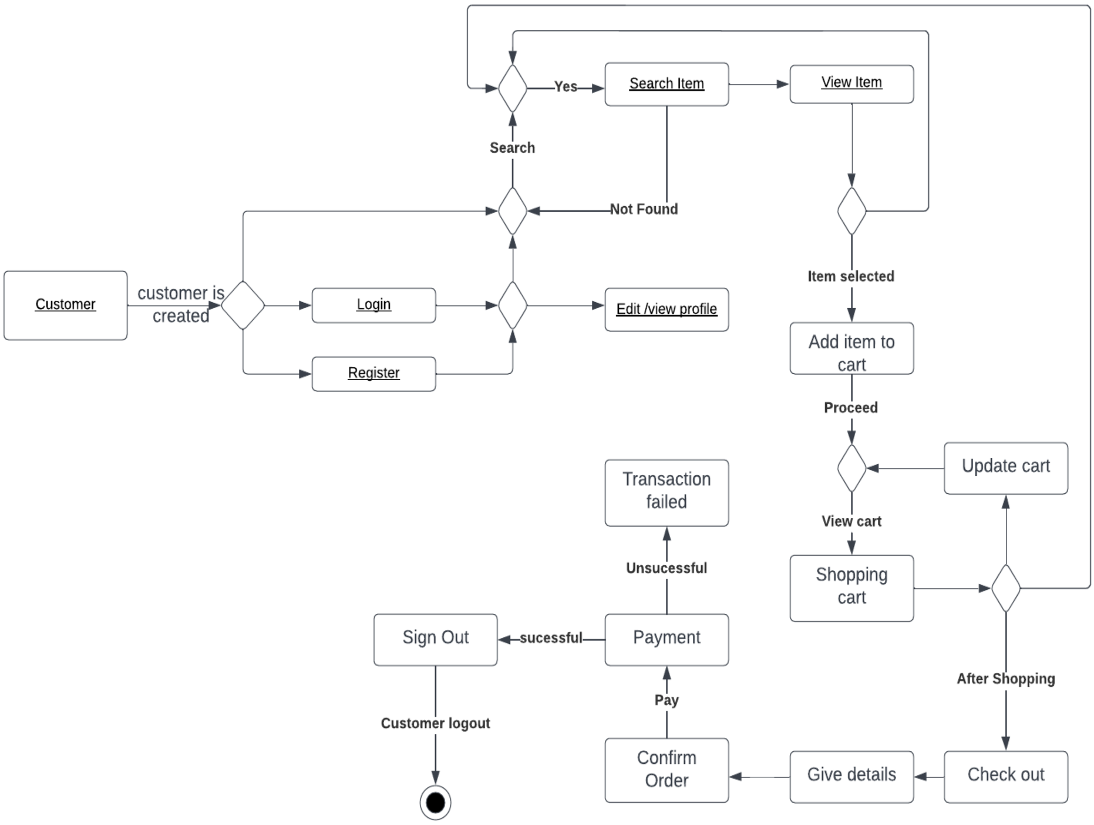
4.2.6 STATECHART DIAGRAM

Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.

Following are the main purposes of using Statechart diagrams –

- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object. Statechart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events.

Fig 6: Statechart diagram for **online plant nursery**



4.2.7 COLLABORATION DIAGRAM

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other.

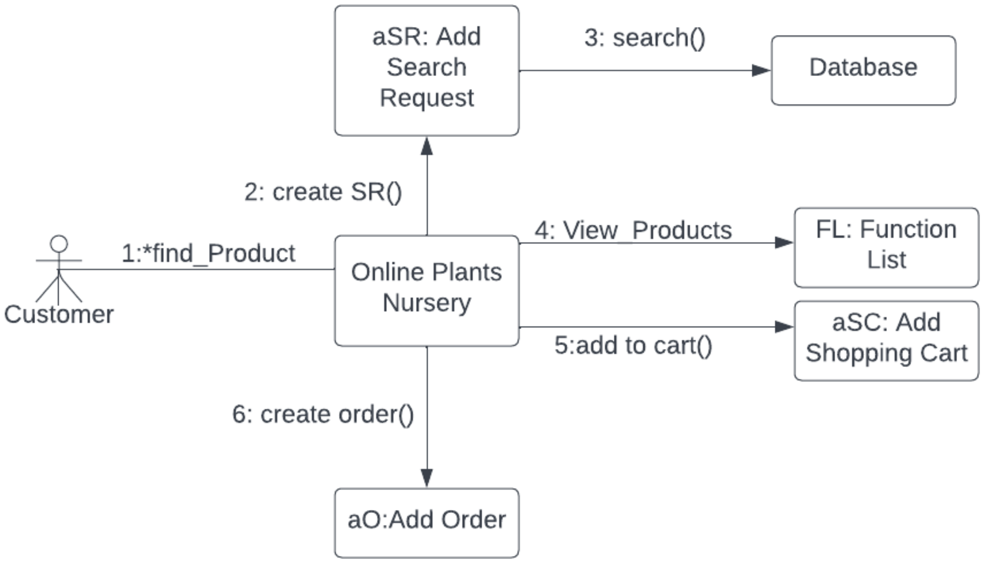
Notations of a Collaboration Diagram,

Objects: The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways: The object is represented by specifying their name and class. It is not mandatory for every class to appear. A class may constitute more than one object.

Actors: In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case. Links:

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line.
Messages: It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

Fig 7: Collaboration diagram for **online plant nursery**



4.2.8 DEPLOYMENT DIAGRAM

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, were software components are deployed. Component diagrams and deployment diagrams are loosely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

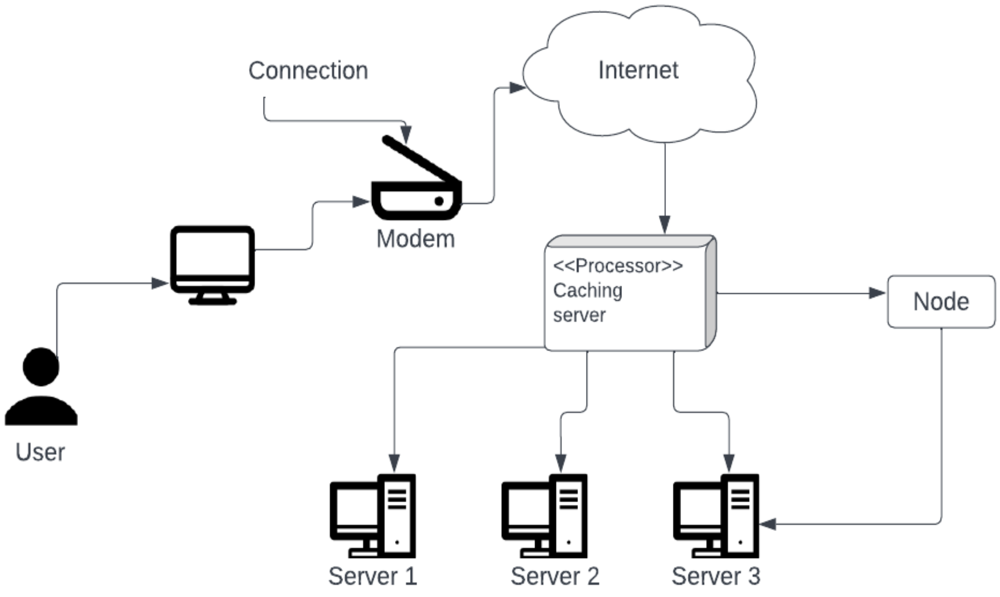
The purpose of deployment diagrams can be described as –

- Visualize the hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe the runtime processing nodes.

An efficient deployment diagram is very important as it controls the following parameters –

- Performance
- Scalability
- Maintainability
- Portability

Fig 8: Deployment diagram for **online plant nursery**

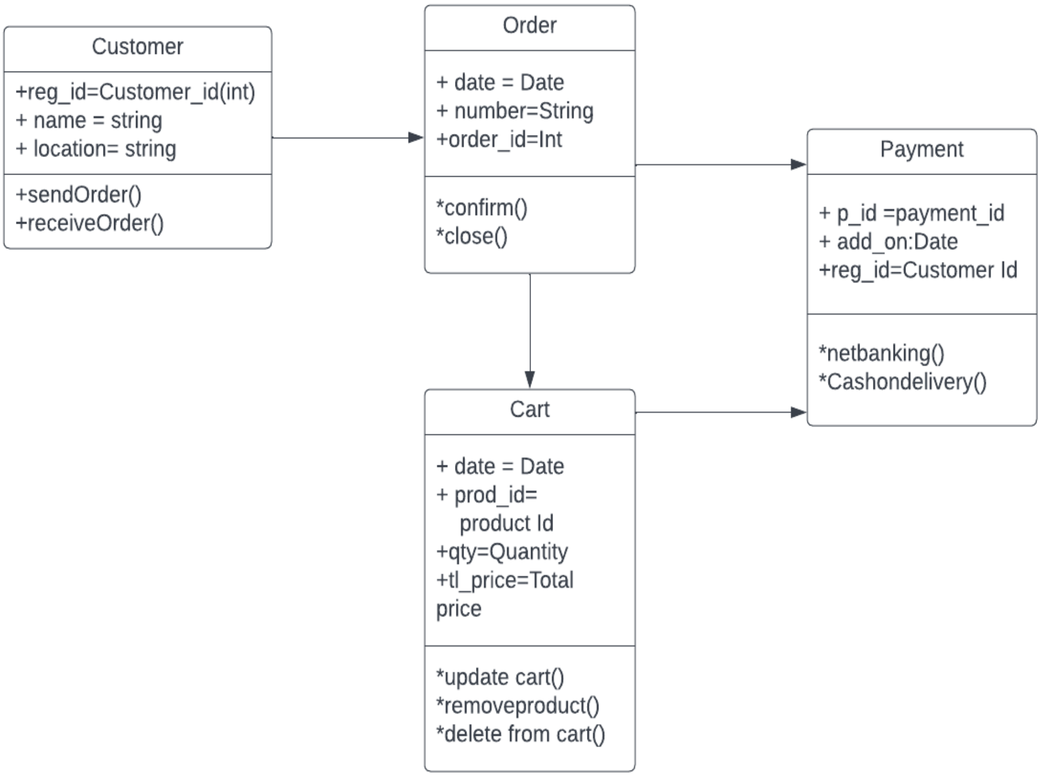


4.2.9 OBJECT DIAGRAM

The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams. The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature. It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment. The purpose of the object diagram can be summarized as –

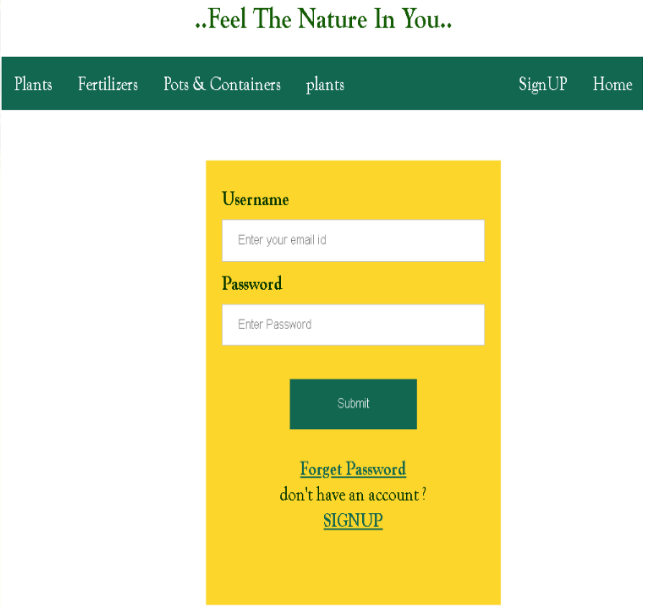
- Forward and reverse engineering.
- Object relationships of a system
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective

Fig 8: Object diagram for **online plant nursery**



4.3 USER INTERFACE DESIGN USING FIGMA

User Login



..Feel The Nature In You..

Username

Enter your email id

Password

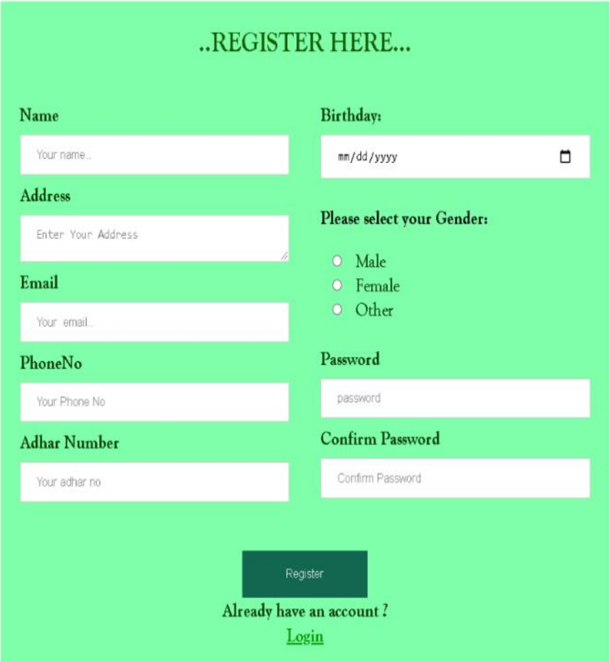
Enter Password

Forget Password

don't have an account ?

SIGNUP

User Register



..REGISTER HERE...

Name

Your name..

Address

Enter Your Address

Email

Your email..

PhoneNo

Your Phone No

Adhar Number

Your adhar no

Birthday:

mm/dd/yyyy

Please select your Gender:

○ Male
○ Female
○ Other

Password

password

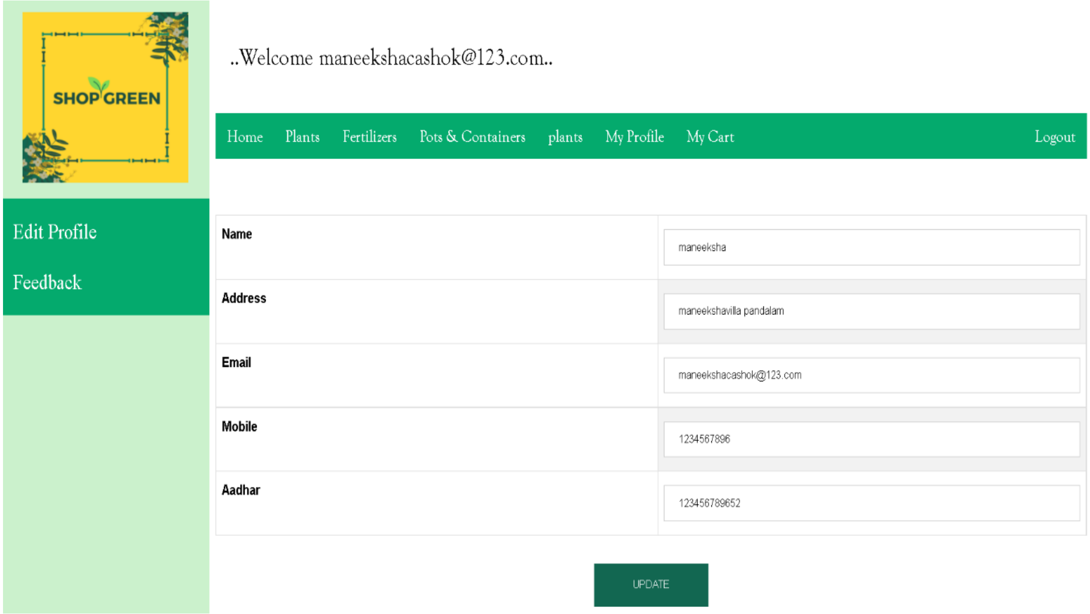
Confirm Password

Confirm Password

Already have an account ?

Login

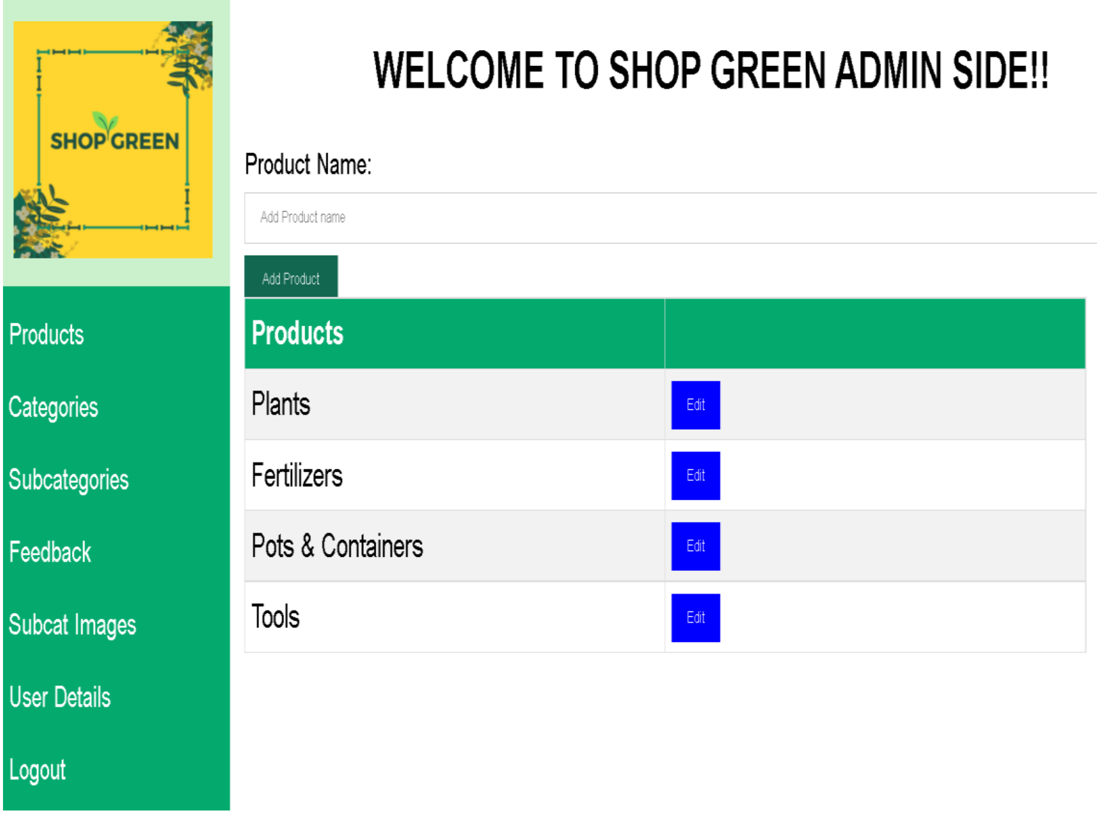
User Update Profile



..Welcome maneekshacashok@123.com..

Name	maneeksha
Address	maneekshavilla pandalam
Email	maneekshacashok@123.com
Mobile	1234567896
Aadhar	123456789652

Add Products

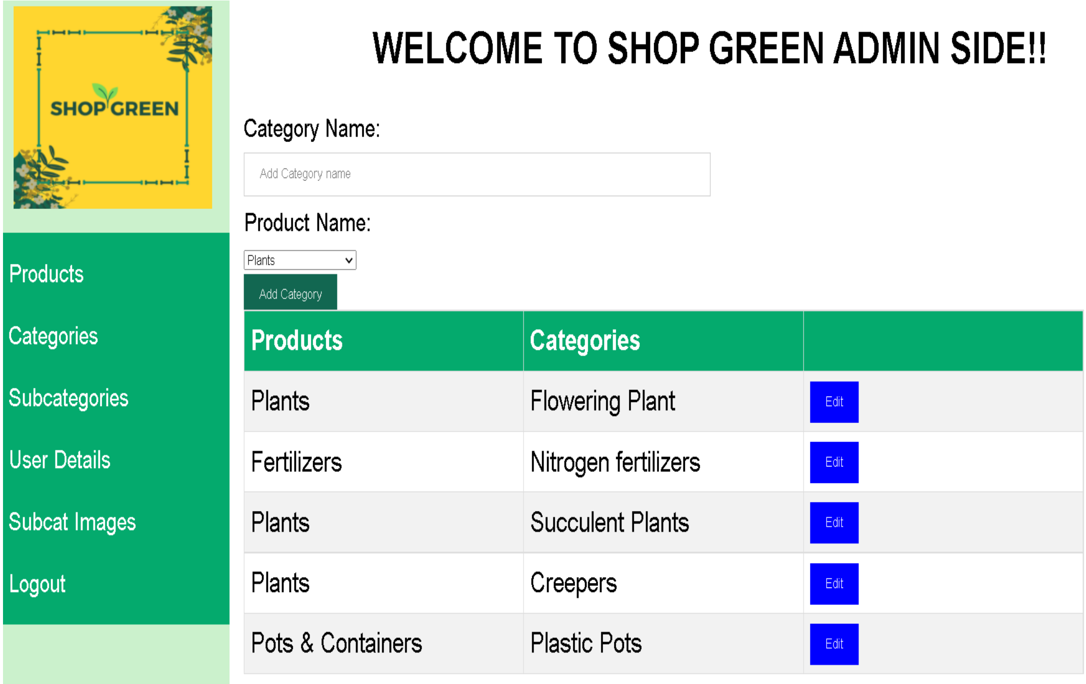


WELCOME TO SHOP GREEN ADMIN SIDE!!

Product Name:

Products	
Plants	Edit
Fertilizers	Edit
Pots & Containers	Edit
Tools	Edit

Add Category



WELCOME TO SHOP GREEN ADMIN SIDE!!

Category Name:

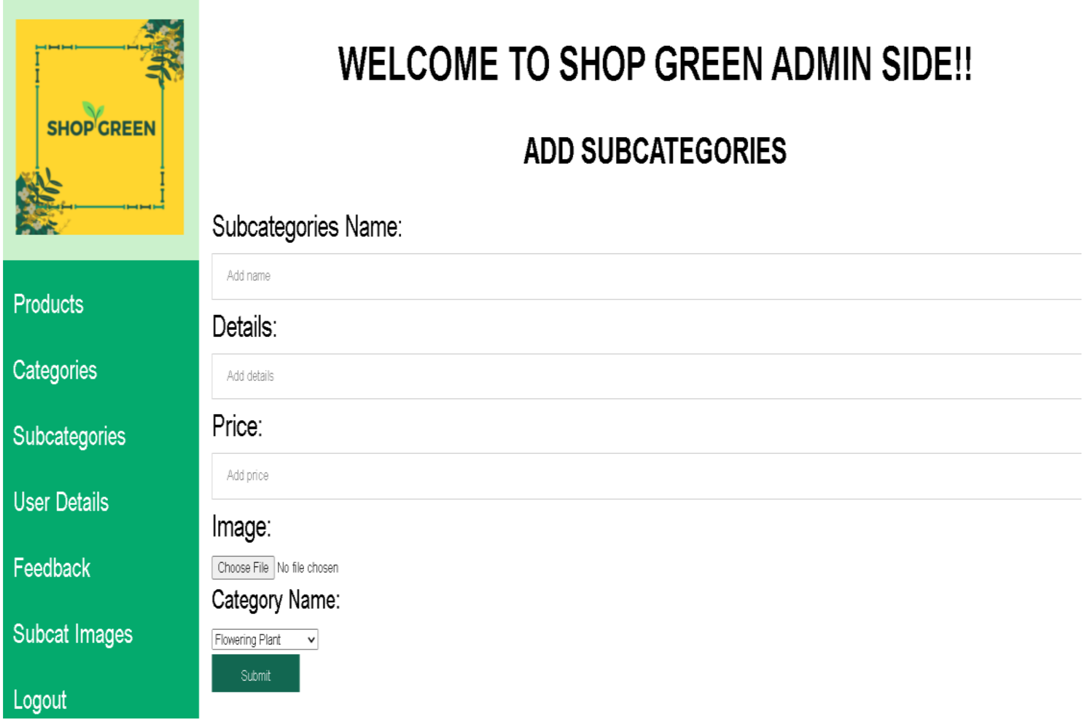
Add Category name

Product Name:

Plants

Products	Categories	
Plants	Flowering Plant	Edit
Fertilizers	Nitrogen fertilizers	Edit
Plants	Succulent Plants	Edit
Plants	Creepers	Edit
Pots & Containers	Plastic Pots	Edit

Add Subcategory



WELCOME TO SHOP GREEN ADMIN SIDE!!

ADD SUBCATEGORIES

Subcategories Name:

Add name

Details:

Add details

Price:

Add price

Image:

No file chosen

Category Name:

Flowering Plant

4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

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 keys are Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ 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 donor 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 attribute 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 another non-key attribute.

TABLE DESIGN

Table No 01: TBL_REGISTER

Primary key: REG_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Reg_id	Int	10	Primary key
Reg_name	Varchar	30	Name of the user.
Reg_email	Varchar	30	Email of the user
Reg_address	Varchar	100	Address
Reg_phone	Int	12	Phone number
Reg_aadhar	Int	12	Aadhar number
Reg_date	Varchar	10	Date of Birth
Reg_gender	Varchar	20	Gender of the user
Password	Varchar	30	Password
Status	Int	2	Availability of user

Table No 02: TBL_User_type

Primary key: Type_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Type_id	Int	10	Primary key
Type_name	Varchar	30	Type of the users

Table No 03: TBL_Login

Primary key: Type_ID

Foreign key: Reg_id, Type_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Login_id	Int	10	Primary key
Reg_id	Int	10	Foreign Key
Type_id	Int	10	Foreign Key

Table No 04: TBL_Product

Primary Key: Prod_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Prod_id	Int	10	Primary key
Prod_name	Varchar	30	Name of Product.

Table No 05: TBL_Category

Primary Key: Cat_ID

Foreign key: Prod_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Cat_id	Int	10	Primary key
Prod_id	Int	10	Foreign key
Cat_name	Varchar	30	Name of Product.

Table No 06: TBL_Category

Primary Key: Cat_ID

Foreign key: Cat_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Sub_Cat_id	Int	10	Primary key
Cat_id	Int	10	Foreign key
Sub_Cat_name	Varchar	10	Name of Product.
Sub_Cat_desc	Varchar	300	Description
Sub_Cat_image	Varchar	255	Image of product
Sub_Cat_price	Varchar	6	Price of product
Sub_Cat_ quantity	Varchar	10	Quantity

Table No 07: TBL_Feedback

Primary Key: F_ID

Foreign key: Reg_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
F_ID	Int	10	Primary Key
Reg_ID	Int	10	Foreign Key
Subject	Varchar	100	Feedback subject
Description	Varchar	300	Description about the feedback

Table No 08: TBL_Cart

Primary Key: Cart_ID

Foreign key: Reg_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Cat_ID	Int	10	Primary Key
Reg_ID	Int	10	Foreign Key
Prod_name	Varchar	100	Name of the Product
Price	Varchar	10	Price of the product
Image	Varchar	300	Image of product
Quantity	Varchar	10	Quantity of the product

Table No 09: TBL_Payment

Primary Key: P_ID

Foreign key: Reg_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
P_ID	Int	10	Primary Key
Reg_ID	Int	10	Foreign Key
Amount	Varchar	100	Amount Paid
Payment Status	Varchar	100	Status of Payment
Payment_Id	Varchar	100	Payment Id
Added On	Varchar	100	Date on which payment is made

Table No 09: TBL_Order

Primary Key: Order_ID

Foreign key: Reg_ID

Foreign key: Subcat_ID

FIELD NAME	TYPE	SIZE	DESCRIPTION
Order_ID	Int	10	Primary Key
Reg_ID	Int	10	Foreign Key
Subcat_ID	Int	100	Foreign Key
Added On	Varchar	100	Date on which payment is made

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- 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 appears to be working according to the specification, that performance requirement appears to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

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 up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

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. Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance 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:

- Input Screen Designs,
- Output 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 uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.2.5 Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software does exactly what it was designed to do. It tests for bugs, defects, and any other issues that can arise with product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs.

Benefits of Automation Testing

- Detailed reporting capabilities - Automation testing uses well-crafted test cases for various scenarios. These scripted sequences can be incredibly in-depth, and provide detailed reports that simply wouldn’t be possible when done by a human.
- Improved bug detection - One of the main reasons to test a product is to detect bugs and other defects. Automation testing makes this process an easier one. It’s also able to analyze a wider test coverage than humans may be able to.
- Simplifies testing - Testing is a routine part of the operations of most SaaS and tech companies. Making it as simple as possible is key. Using automation is extremely beneficial. When automating test tools, the test scripts can be reused
- Speeds up the testing process - Machines and automated technology work faster than humans. Along with improved accuracy, this is why we use them. In turn, this shortens your software development cycles.
- Reduces human intervention - Tests can be run at any time of day, even overnight, without needing humans to oversee it. Plus, when it’s conducted automatically, this can also reduce the risk of human error

5.2.6Selenium Testing

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, and C#, among others. The Selenium testing tool is used to automate tests across browsers for web applications. It's used to ensure high-quality web applications — whether they are responsive, progressive, or regular. Selenium is an open-source tool.

Test cases for a Login Page

Project Name: Online Plants Nursery	
Login Test Case	
Test Case ID: Fun_1	**Test Designed By: Maneeksha C Ashok**
Test Priority (Low/Medium/High): High	**Test Designed Date:** 22-05-2022
Module Name: Login Screen	**Test Executed By: Ms. Shelly Shiju George**
Test Title: Verify login with valid email and password	**Test Execution Date:** 23-05-2022
Description: Test the Login Page	
Pre-Condition: User has valid email id and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login Page should be display ed	Login page displayed	Pass
2	Provide Valid Email Id	maneeksha@123.com	User should d be able to Login	User Logged in and navigated to User Dashboard with records	Pass
3	Provide Valid Password	Password:123			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: maneeksha@123. com Password: User12345	User should not be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null Email Id or Password	Email Id: null Password: null			
7	Click on Sign In button				

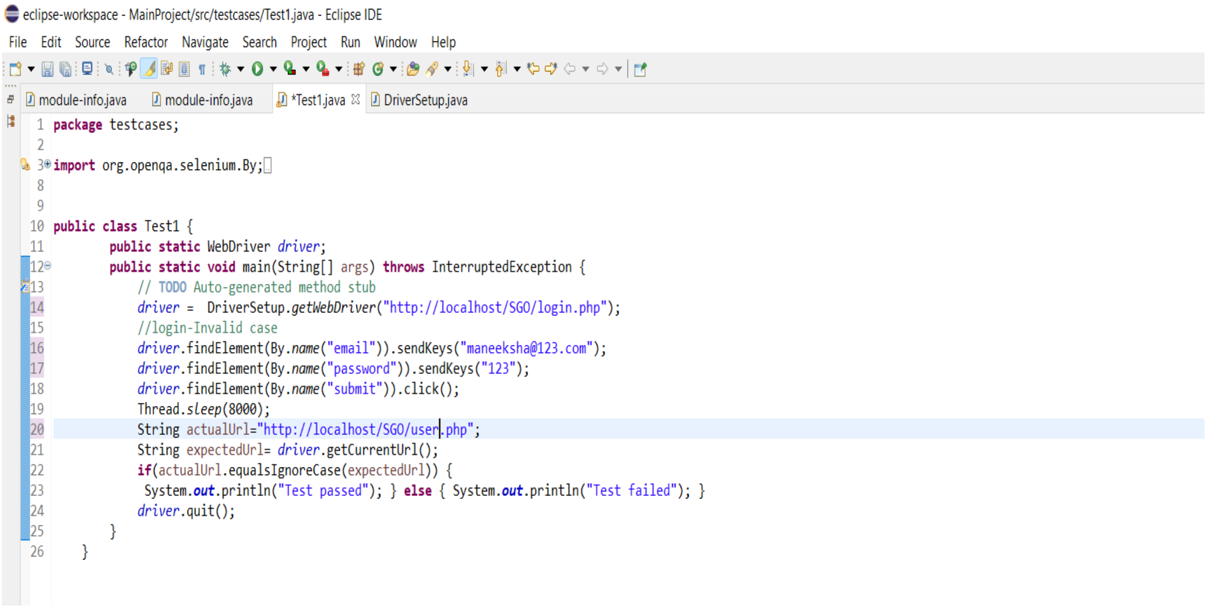
Post-Condition: User is validated with database and successfully login into account.

The Account session details are logged in database.

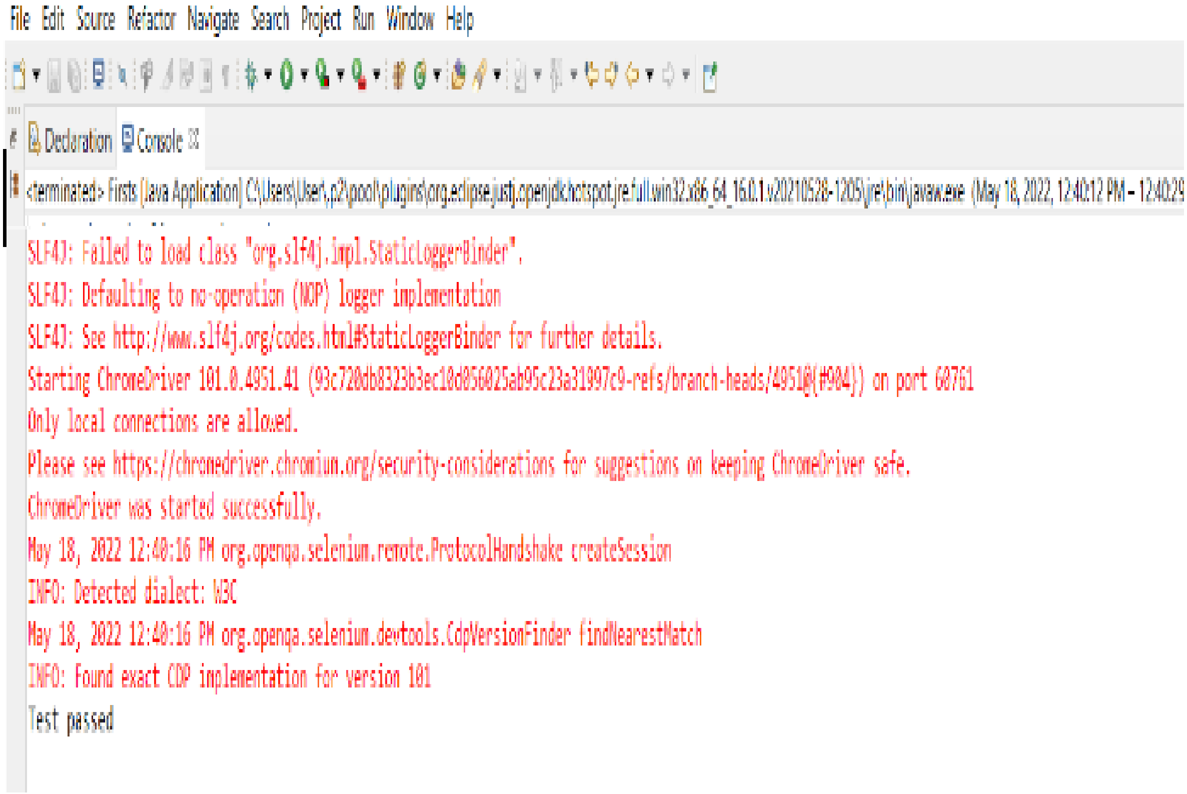
Code package

```
testcase; import org. openqa. selenium.By;
 import org. openqa. selenium.WebDriver;
import browserimplement.DriverSetup;
 public class Firsts {
 public static WebDriver driver;
public static void main (String [] args) throws InterruptedException {
 // TODO Auto-generated method stub driver = DriverSetup.getWebDriver("http://localhost/SGO/
Login.php");
//login-Invalid case driver. findElement(By.name("email")).sendKeys("maneeksha@123.com");
driver.findElement(By.name("psw")).sendKeys("123");
driver.findElement(By.name("submit")).click(); Thread.sleep(8000);
 String actualUrl="http://localhost/SGO/ User.php";
String expectedUrl= driver.getCurrentUrl();
 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
 System.out.println("Test passed");
 } else
 {
 System.out.println("Test failed");
 }
driver.quit();
} }
```

Output


eclipse-workspace - MainProject/src/testcases/Test1.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
module-info.java
module-info.java
*Test1.java
DriverSetup.java

```java
package testcases;

import org.openqa.selenium.By;


public class Test1 {
        public static WebDriver driver;
        public static void main(String[] args) throws InterruptedException {
            // TODO Auto-generated method stub
            driver =  DriverSetup.getWebDriver("http://localhost/SGO/login.php");
            //login-Invalid case
            driver.findElement(By.name("email")).sendKeys("maneeksha@123.com");
            driver.findElement(By.name("password")).sendKeys("123");
            driver.findElement(By.name("submit")).click();
            Thread.sleep(8000);
            String actualUrl="http://localhost/SGO/user.php";
            String expectedUrl= driver.getCurrentUrl();
            if(actualUrl.equalsIgnoreCase(expectedUrl)) {
             System.out.println("Test passed"); } else { System.out.println("Test failed"); }
            driver.quit();
        }
    }
```


File Edit Source Refactor Navigate Search Project Run Window Help
Declaration
Console

```
<terminated> Firsts [Java Application] C:\Users\User\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (May 18, 2022, 12:40:12 PM – 12:40:29
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 60761
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 18, 2022 12:40:16 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 18, 2022 12:40:16 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

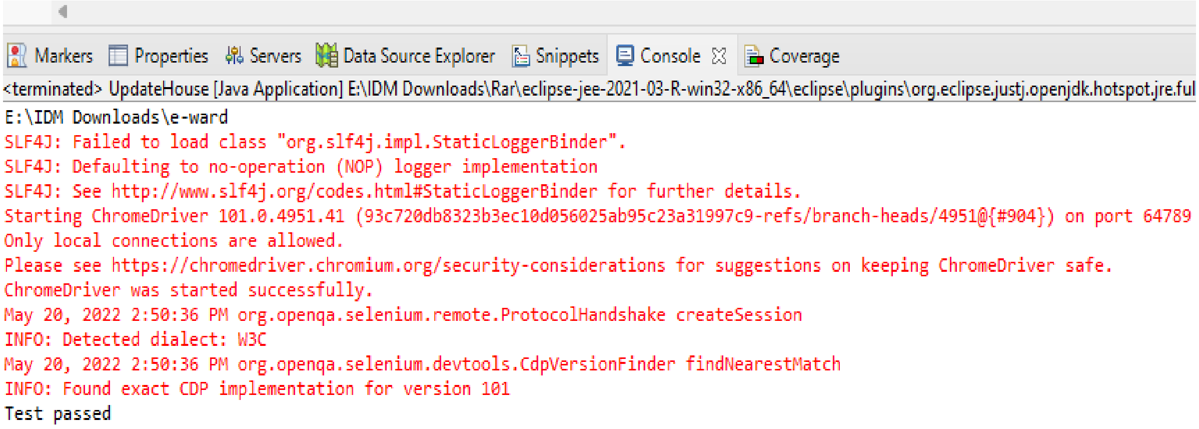
Test cases for Update Profile

Project Name: Online Plants Nursery					
Updation Test Case					
Test Case ID: update Profile			**Test Designed By: Maneeksha C Ashok**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 23-05-2022		
Module Name: Login Screen			**Test Executed By: Ms. Shelly Shiju George**		
Test Title: Update Profile Details			**Test Execution Date:** 24-05-2022		
Description: Login to system and update profile information, 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 Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid Username	User Name: maneeksha @123.com	User should be able to Login	User Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: 123			
4	Click on Login button				
5	Provide profile details	Input profile details	User will be redirected to dashboard	User will be redirected to dashboard	Pass
7	Click on Update button				
8	Provide invalid information	Input invalid profile details.	User will be redirected to dashboard	User will be stay on that page showing error message	Pass
9	Click on Update button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

Code

```
package testcases;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.DriverSetup;

public class UpdateProfile {
public static WebDriver driver;
public static void main(String[] args) {
driver =  DriverSetup.getWebDriver("http://localhost/SGO/ index.php ");
driver.findElement(By.name("login")).click();
driver.findElement(By.name("userName")).sendKeys("maneeksha@123.com");
driver.findElement(By.name("password")).sendKeys("123");
driver.findElement(By.name("submitButton")).click();
driver.get("http://localhost/SGO/ profile.php");

driver.findElement(By.name("address")).sendKeys("Thamburu Thonnalloor  Pandalam");
driver.findElement(By.name("email")).sendKeys("maneeksha@123.com");
driver.findElement(By.name("phone")).sendKeys("8606345393");
driver.findElement(By.name("uphbtn")).click();

String actualUrl="http://localhost/SGO/ profile.php";
String expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
```

Output



```
<terminated> UpdateHouse [Java Application] E:\IDM Downloads\Rar\eclipse-jee-2021-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.ful
E:\IDM Downloads\e-ward
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 64789
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 20, 2022 2:50:36 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 20, 2022 2:50:36 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

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 the software but we have to ensure that the resistance does not build up, 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.

User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system.

As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

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 date 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

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.

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"

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The project entitled Online Plant Nursery was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for purchasing Gardening Items and Plants Online. This project helped in gaining valuable information and practical knowledge on several topics like designing web pages using html &CSS, usage of responsive templates, and management of database using MySQL. The entire system is secured. Also, the project helped in understanding about the development phases of a project and software development life cycle. It helps to learn how to test different features of a project. This project has given a great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops to sell various kinds of products by simple modifications. There is a scope for further development in this project to a great extent.

7.2 FUTURE SCOPE

- Staffs can be added.
- Newsletter subscription can be added.
- Customer portal can be added to sell their own plants.
- Communication through chats.
- Transferring data with high security
- Data security can be enhanced.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Jon Duckett., Java Script Programming, Wiley, 2014
- Bryan Basham, Kathy Sierra., Head First EJB, O’Reilly Media; Second edition, 2008
- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- PankajJalote, “Software engineering: a precise approach”, 2006.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- https://www.tutorialspoint.com/index.htm
- https://www.javatpoint.com
- https://www.w3schools.com
- https://html.com

CHAPTER 9

APPENDIX

8.1 SAMPLE CODE

addssubcategory.php

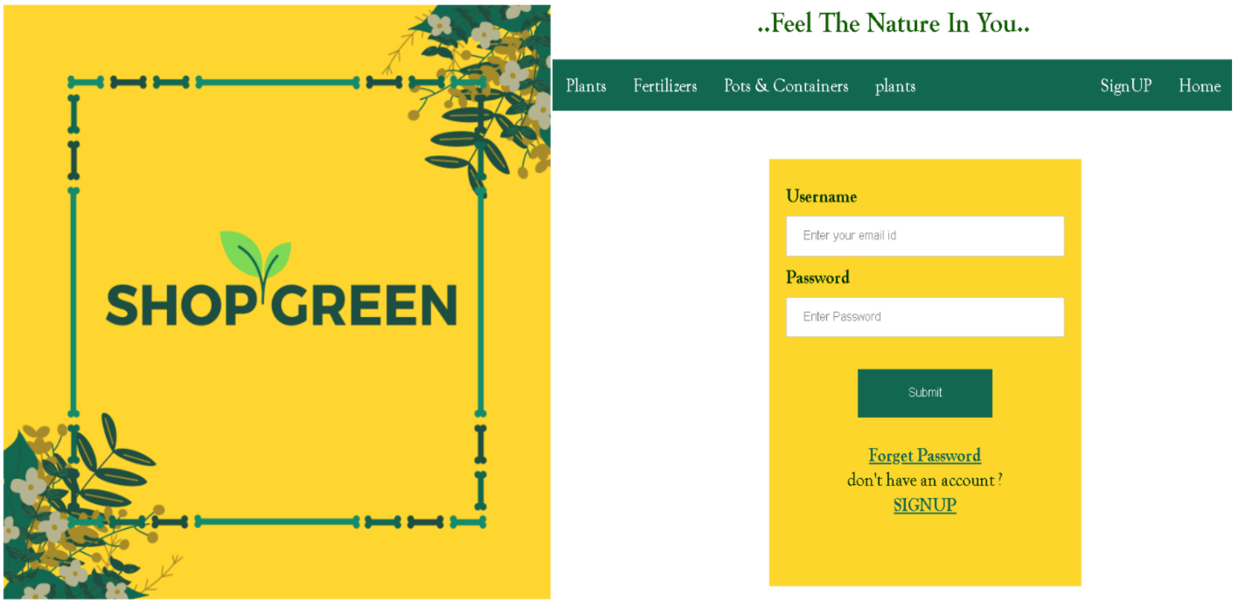
```
<?php
include('configr.php');
if(isset($_POST['subm']))
{
   $a=$_POST['cat'];
   $b=$_POST['sub'];

$query="SELECT * FROM `product` WHERE prod_name='$b' ";
$data = mysqli_query($con,$query);
if($res=mysqli_fetch_assoc($data))
{
 $ker=$res['prod_id'];
 //echo $ker;
$sql="INSERT INTO `category`( `prod_id`, `cat_name`) VALUES ('$ker','$a')";
 if(mysqli_query($con,$sql))
 {
             echo "registered successfully";

     }
     else
     {
             echo mysqli_errno($con);
     }       }
}
?>
<DOCTYPE! HTML>
<html>
<head>
<link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
<style>
input[type=submit] {

  background-color:#126751;

color: white;
```

```
 border: 1px;
 border-color:green;
 cursor: pointer;
 float:center;
 width:120px;
 height:35px;
}
input[type=text]
 {
       width: 50%;
       padding: 12px 20px;
       margin: 8px 0;
       display: inline-block;
       border: 1px solid #ccc;
       color:black;
           float:center;
   }
 optgroup { font-size:20px; }
</style>
</head>
<body>
<div class="pannel">
 <div class="rimg"><img src="images/3.png"  height="100"  align="left">
 <p></p>
</div>
<div class="navbar">
 <a href="adminhome.php">Home</a>
 <div class="dropdown">
   <button class="dropbtn">View
     <i class="fa fa-caret-down"></i>
   </button>
   <div class="dropdown-content">
     <a href="usrdtil.php">userdetails</a>
     <a href="bokdtil.php">bookingdetails</a>
     <a href="dcontct.php">contact details</a>
    <a href="dstaff.php">staff details</a>
      </div>
 </div>

      <div class="dropdown">
  <button class="dropbtn">Add Products
    <i class="fa fa-caret-down"></i>
  </button>
  <div class="dropdown-content">
    <a href="addcat.php">Categories</a>
    <a href="addsubc.php">Sub-categories</a>
  <a href="addplants.php">Products</a>
</div>
</div>
</div>
<div class="fort" >
 <form method="POST" action="addsubc.php">
<label style="font-size:25px">Category Name</label><br>
  <input type="text" id="sub" name="cat" placeholder="Add  name" required><br>
 <label style="font-size:25px">Product Name:</label><br>
  <select required name="sub" id="sub">
 <optgroup>
  <?php
       include('configr.php');

       $query="SELECT * FROM `product`";

$data = mysqli_query($con,$query);
 while($res=mysqli_fetch_assoc($data))
 {
      ?>
  <option><?php echo $res['prod_name'];?></option>
 <?php
 }
 ?>   </optgroup></select><br>
          <br><input type="submit"name="subm" value="Submit"><br><br></form>
    <br>
    <table border="4px"   style="border-color:firebrick;width:90% ;font-size:25px ;margin-left:0%">
 <tr>  </td>Product</td><td>Category</td>
 </tr> <?php
                include('configr.php')
                          $query=" SELECT product.prod_name ,category.cat_name
FROM product JOIN category
ON (product.prod_id = category.prod_id)";
$data = mysqli_query($con,$query);
    while($res=mysqli_fetch_assoc($data))
    {
              ?>
        <tr><td><?php echo $res['prod_name'];?></td>
              <td><?php echo $res['cat_name'];?></td>
              </tr>
    <?php
    }
    ?>
</table>
</div>
</body>
</html>
<script type = "text/javascript" >
    function preventBack() { window.history.forward(); }
    setTimeout("preventBack()", 0);
    window.onunload = function () { null };
</script>
```

9.2 SCREEN SHOTS

CUSTOMER PAGES

User Login



..Feel The Nature In You..

Username

Enter your email id

Password

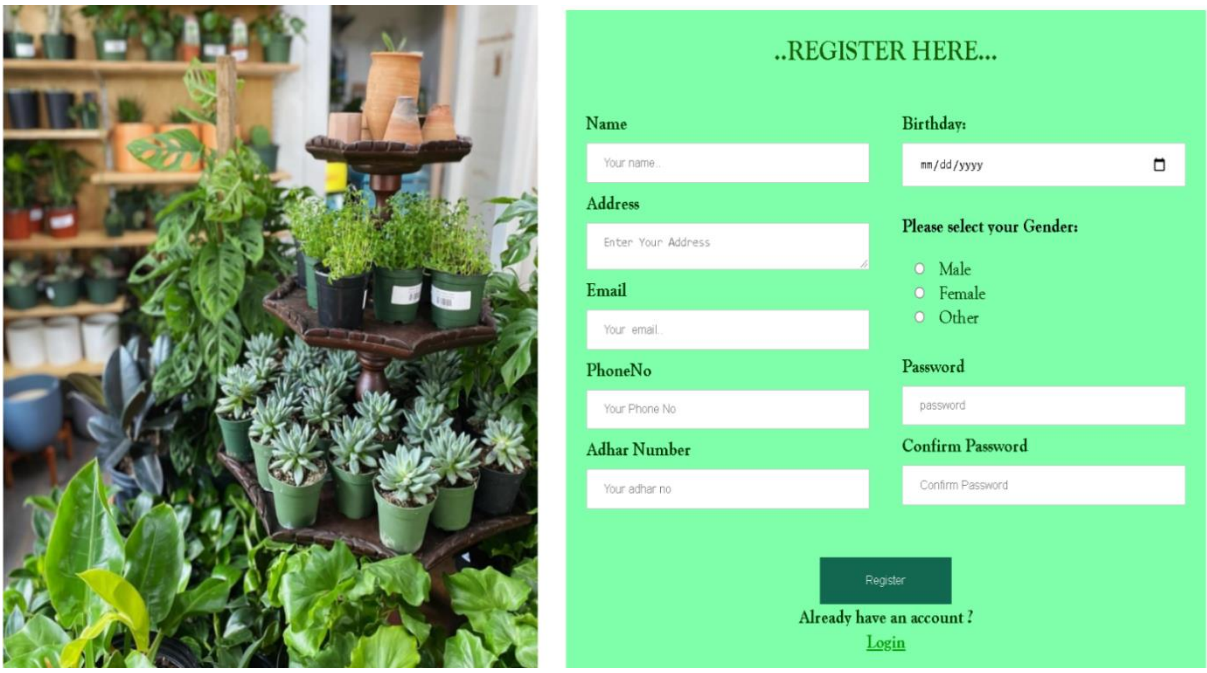
Enter Password

Forget Password

don't have an account ?

SIGNUP

User Register



..REGISTER HERE...

Name

Your name..

Address

Enter Your Address

Email

Your email..

PhoneNo

Your Phone No

Adhar Number

Your adhar no

Birthday:

mm/dd/yyyy

Please select your Gender:

○ Male

○ Female

○ Other

Password

password

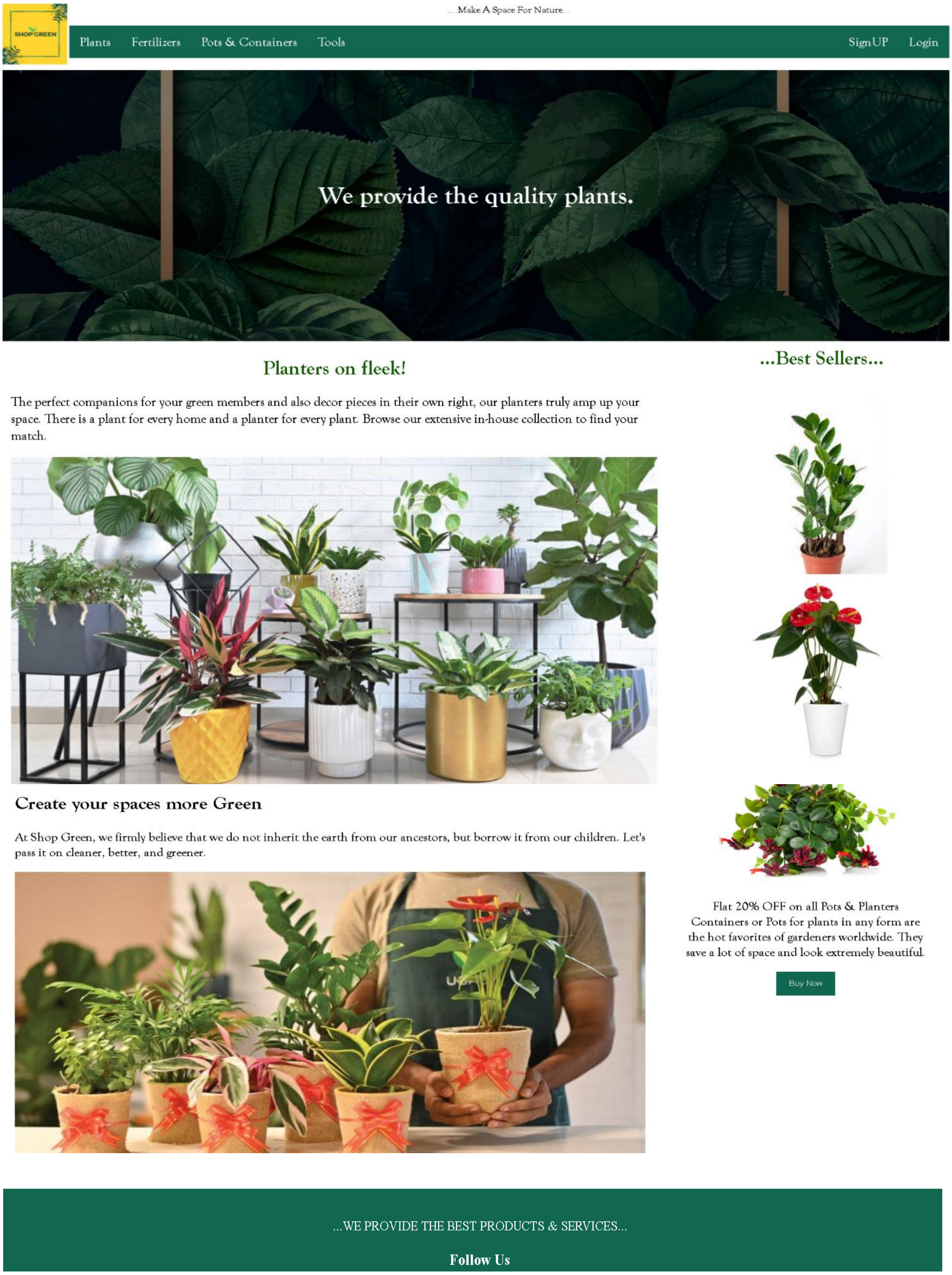
Confirm Password

Confirm Password

Already have an account ?

Login

Home Page



.....Make A Space For Nature.....

We provide the quality plants.

Planters on fleek!

The perfect companions for your green members and also decor pieces in their own right, our planters truly amp up your space. There is a plant for every home and a planter for every plant. Browse our extensive in-house collection to find your match.

Create your spaces more Green

At Shop Green, we firmly believe that we do not inherit the earth from our ancestors, but borrow it from our children. Let's pass it on cleaner, better, and greener.

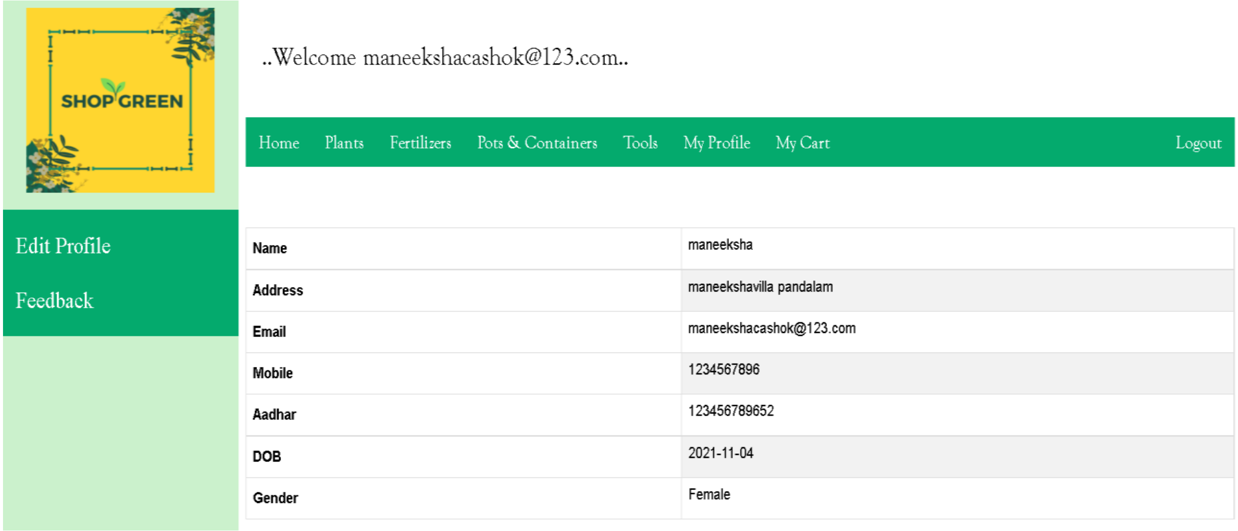
...Best Sellers...

Flat 20% OFF on all Pots & Planters
Containers or Pots for plants in any form are the hot favorites of gardeners worldwide. They save a lot of space and look extremely beautiful.

...WE PROVIDE THE BEST PRODUCTS & SERVICES...

Follow Us

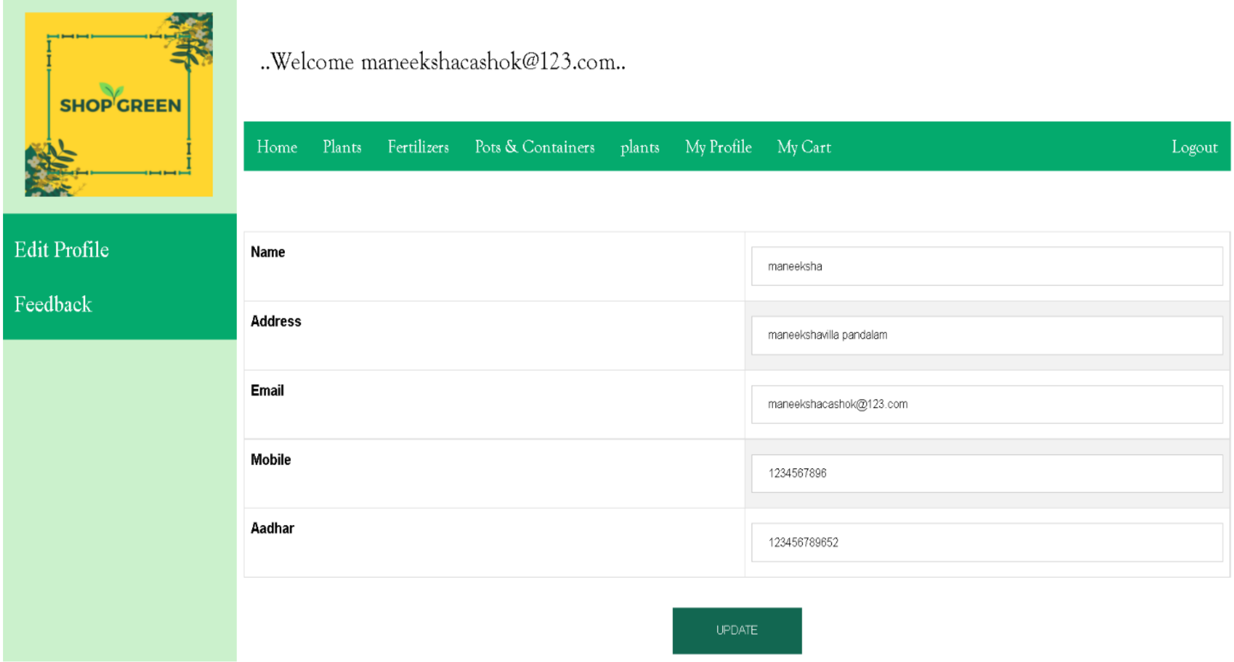
User Dashboard



..Welcome maneekshacashok@123.com..

Name	maneeksha
Address	maneekshavilla pandalam
Email	maneekshacashok@123.com
Mobile	1234567896
Aadhar	123456789652
DOB	2021-11-04
Gender	Female

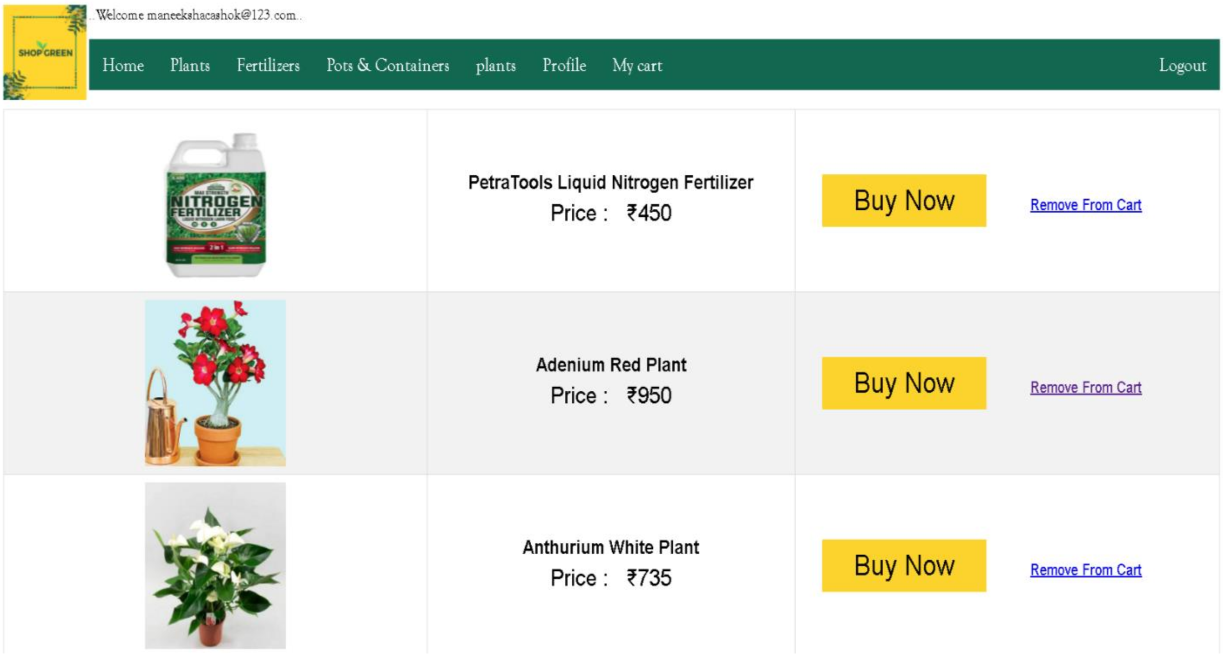
User Update Profile



..Welcome maneekshacashok@123.com..

Name	maneeksha
Address	maneekshavilla pandalam
Email	maneekshacashok@123.com
Mobile	1234567896
Aadhar	123456789652

User Cart

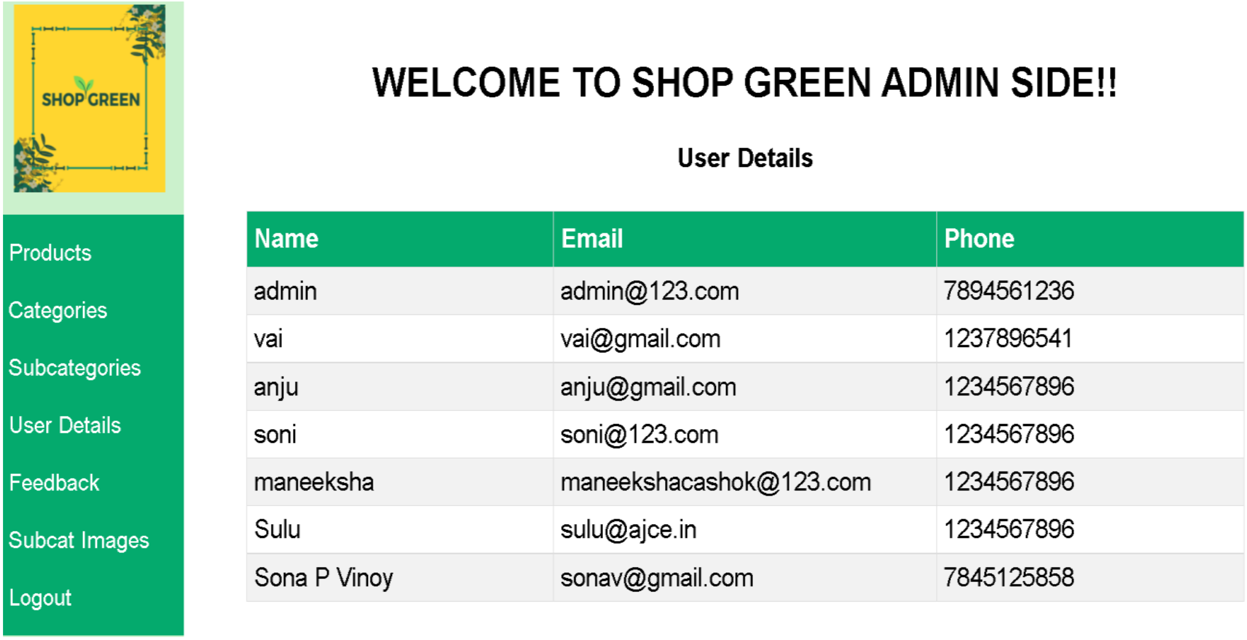


Welcome maneekshacashok@123.com..

	PetraTools Liquid Nitrogen Fertilizer Price : ₹450	Buy Now Remove From Cart
	Adenium Red Plant Price : ₹950	Buy Now Remove From Cart
	Anthurium White Plant Price : ₹735	Buy Now Remove From Cart

ADMIN PAGES

Admin Dashboard

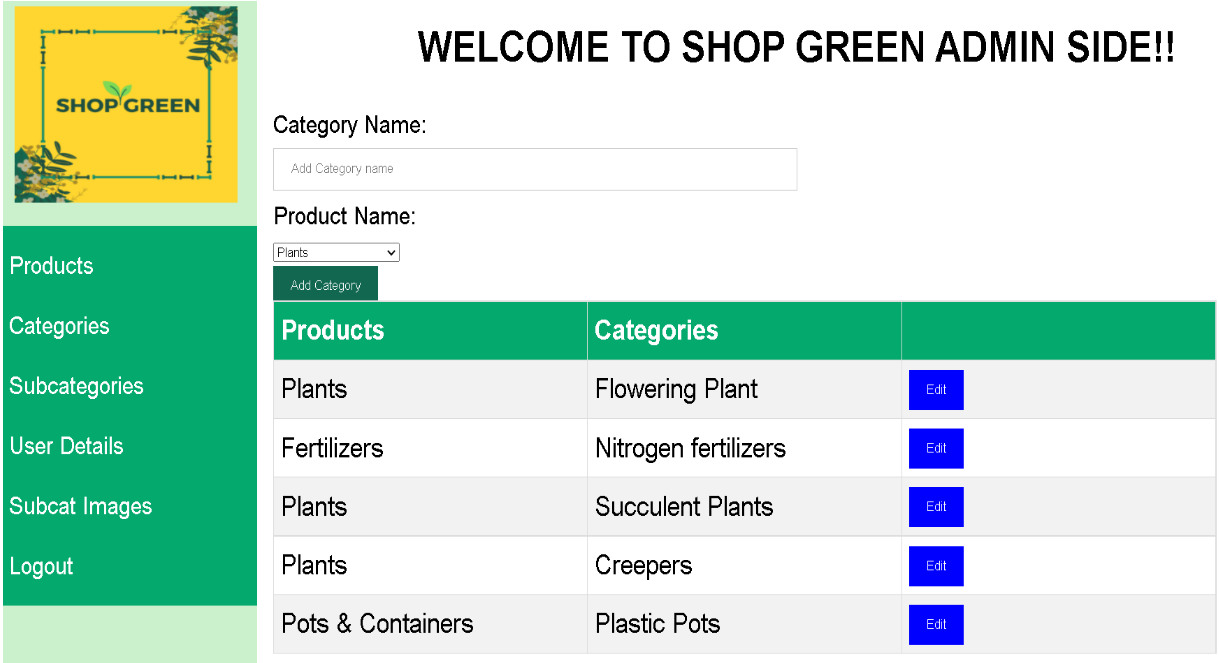


WELCOME TO SHOP GREEN ADMIN SIDE!!

User Details

Name	Email	Phone
admin	admin@123.com	7894561236
vai	vai@gmail.com	1237896541
anju	anju@gmail.com	1234567896
soni	soni@123.com	1234567896
maneeksha	maneekshacashok@123.com	1234567896
Sulu	sulu@ajce.in	1234567896
Sona P Vinoy	sonav@gmail.com	7845125858

Add Products



WELCOME TO SHOP GREEN ADMIN SIDE!!

Category Name:

Add Category name

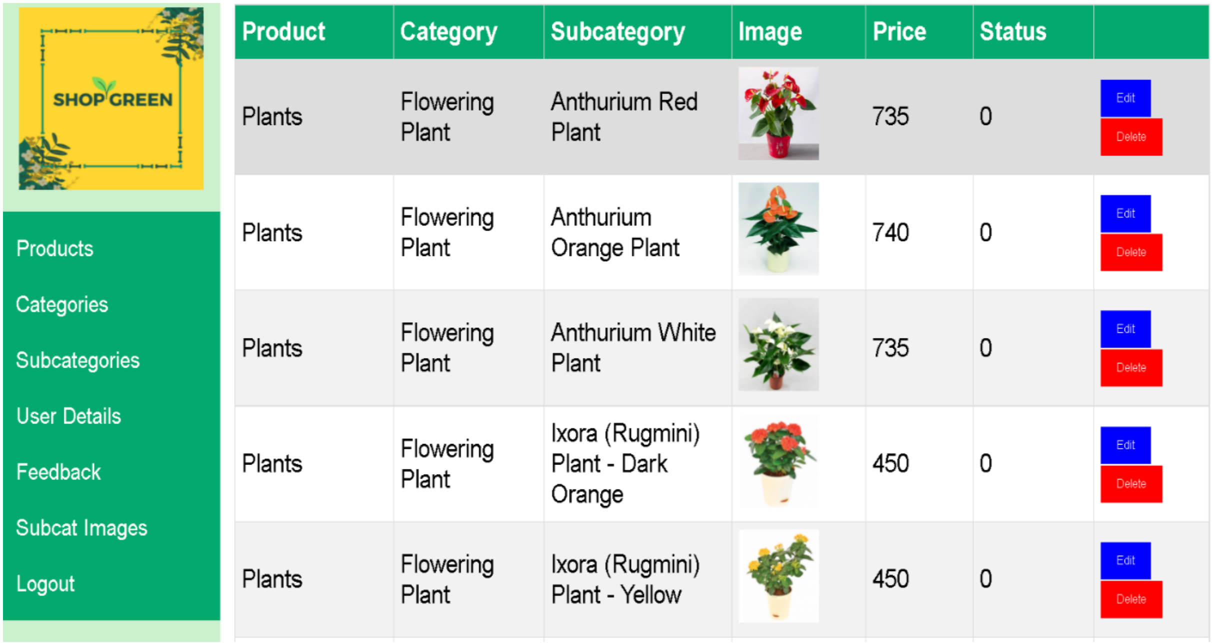
Product Name:

Plants

Add Category

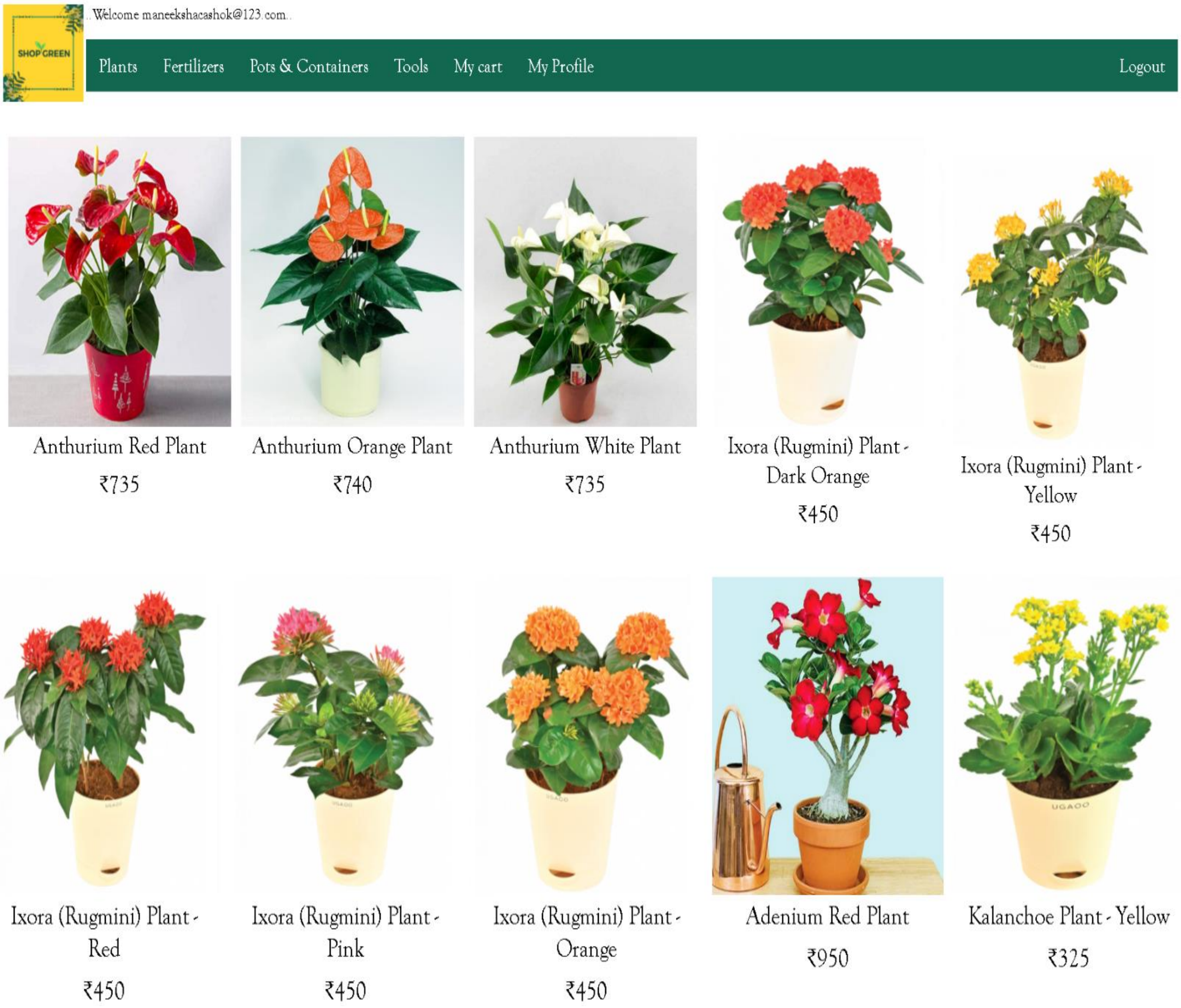
Products	Categories	
Plants	Flowering Plant	Edit
Fertilizers	Nitrogen fertilizers	Edit
Plants	Succulent Plants	Edit
Plants	Creepers	Edit
Pots & Containers	Plastic Pots	Edit

View Products



Product	Category	Subcategory	Image	Price	Status	
Plants	Flowering Plant	Anthurium Red Plant		735	0	Edit Delete
Plants	Flowering Plant	Anthurium Orange Plant		740	0	Edit Delete
Plants	Flowering Plant	Anthurium White Plant		735	0	Edit Delete
Plants	Flowering Plant	Ixora (Rugmini) Plant - Dark Orange		450	0	Edit Delete
Plants	Flowering Plant	Ixora (Rugmini) Plant - Yellow		450	0	Edit Delete

User View of Products



Welcome maneekshacashok@123.com

Anthurium Red Plant
₹735

Anthurium Orange Plant
₹740

Anthurium White Plant
₹735

Ixora (Rugmini) Plant - Dark Orange
₹450

Ixora (Rugmini) Plant - Yellow
₹450

Ixora (Rugmini) Plant - Red
₹450

Ixora (Rugmini) Plant - Pink
₹450

Ixora (Rugmini) Plant - Orange
₹450

Adenium Red Plant
₹950

Kalanchoe Plant - Yellow
₹325