G&S MOTOR-PUMP STORE

Project Report Submitted By

JYOTHIKA SURESH

Reg. No.: AJC17MCA-I026

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

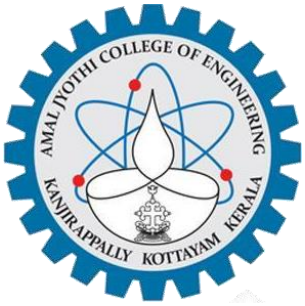
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2017-2022

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, "**G&S MOTOR-PUMP STORE**" is the bonafide work of **JYOTHIKA SURESH (Reg.No: AJC17MCA-I026)** 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 2017-22.

Ms. Meera Rose Mathew
Internal Guide

Rev. Fr. Dr. Rubin Thottupurathu Jose
Coordinator

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

External Examiner

DECLARATION

I hereby declare that the project report **"G&S MOTOR PUMPSTORE"** 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 2017-2022.

Date: **JYOTHIKA SURESH**

KANJIRAPPALLY **Reg. No: AJC17MCA-I026**

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 and the project coordinator **Rev.Fr. Dr. Rubin Thottupurathu Jose** for his 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. Meera Rose Mathew** 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.

JYOTHIKA SURESH

ABSTRACT

G&S Motor Pump Store provides the customers with information on various pump sets, motors etc... under the company KBL, L&T, Honda. This is a site to login to get the latest updates of our products. An easy way to buy a product just sitting in front of your computers by registering into our site. Also, we provide cart so customers can by multiple products at a time.

Our system contains a dashboard page containing product management that is managing category list, sub-category list etc... with an integrated Payment gateway and Bill Generation. And customers get a chance to service their bought product. Also, a staff portal is included such that the staff could see which service is assigned to them.

The proposed system has features that is mentioned above.

Purchase Bill is the receipt received by the customer for the product he has purchased. A bill is generally a document specifying money to be paid against the order placed. Purchase Bill is the receipt received by the customer for the product he has purchased. And the services include oil changing, check-up etc... Also, staff can update their profile and upload required documents.

.

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

This project is a web-based shopping system for an existing shop. The project objective is to deliver an online shop management which is meant to help the customers to make their purchase easy. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. Thus, the customer will get the service of online shopping from his favorite shop.

Our system contains a dashboard page containing product management that is managing category list, sub-category list etc... with an integrated Payment gateway and Bill Generation. And customers get a chance to service their bought product. Also, a staff portal is included such that the staff could see which service is assigned to them.

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can buy a product from the site. Also, the customers can access to the web where he/she can buy a product and corresponding bill is generated. Also that product can be serviced, by sending a service request our staff will be at your doorstep for the service.

The system includes 3 modules. They are:

1. **Admin Module**

Admin must have a login into this system. He has the overall control of the system. Admin can add or update product details, manage user data etc. Admin can View all the registered users and can manage the products.

- Manage Categories and Subcategories.
- Add Products: The website contains different kind of products. The products can be classified into different categories by its type. Admin can add new products into the existing system with all its details including an image.
- Delete Products: Administrator can delete the products based on the stock of that particular product.
- Update Products: The product details can be updated at any time.

2. **Staff Module**

Staff can login to the staff portal by the login credentials provided by the admin. And can set new password and update their profile. Also, they can upload various documents that are needed. In the

staff portal they could see, to which customer they are assigned for service.

3. **Customer Module**

Customer can register into the site to discover more products and can add that product to cart and

also view information about the product.

a. Not a registered a credentials:

They are only allowed to surf through the system, and ask any question or any queries. They are limited to access for the site full potentials. Moreover, functionalities like buying the product is only for registered users.

b. Registered credentials:

They are real users of the system and they are provided with the functionalities of buying the product and see the full detailed product information.

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.

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.

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 a complete online shop. Customer can register and they can see all the product of each category and can add the product of their wish to the cart. Each customer can create their own profile. The proposed system rectifies the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- No proper online management of system
- No bill generation.
- Order management is not done.
- Human effort is needed.

2.4 PROPOSED SYSTEM

The proposed system has features that is mentioned above. Purchase Bill is the receipt received by the customer for the product he has purchased. A bill is generally a document specifying money to be paid against the order placed. Product comparison makes you select the better product out of two similar products. Q&A section includes the answers to common queries. Also, latest news or updates from the company also will be presented on the site. Monthly sales report helps to know which product is sold most, earned profit etc.…

2.5 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:

- **Better security: -**

  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.

- **Ensure data accuracy: -**

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

- **Better service: -**

  The product 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

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 Economic 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, G&S Motor Pump Store 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.

G&S MOTOR PUMP STORE, GUI is simple so that users can easily use it also it 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, AJAX, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 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. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP 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.

3.3.2 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 and several versions exist. In this manual, "SQL92" refers to the standard released in 1992,

  "SQL: 1999" refers to the standard released in 1999, and "SQL: 2003" refers to the current version of the standard. We use the phrase "the SQL standard" to mean the current version of the SQL Standard at any time.

- **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
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component 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 for the modeling of real-world objects and systems.

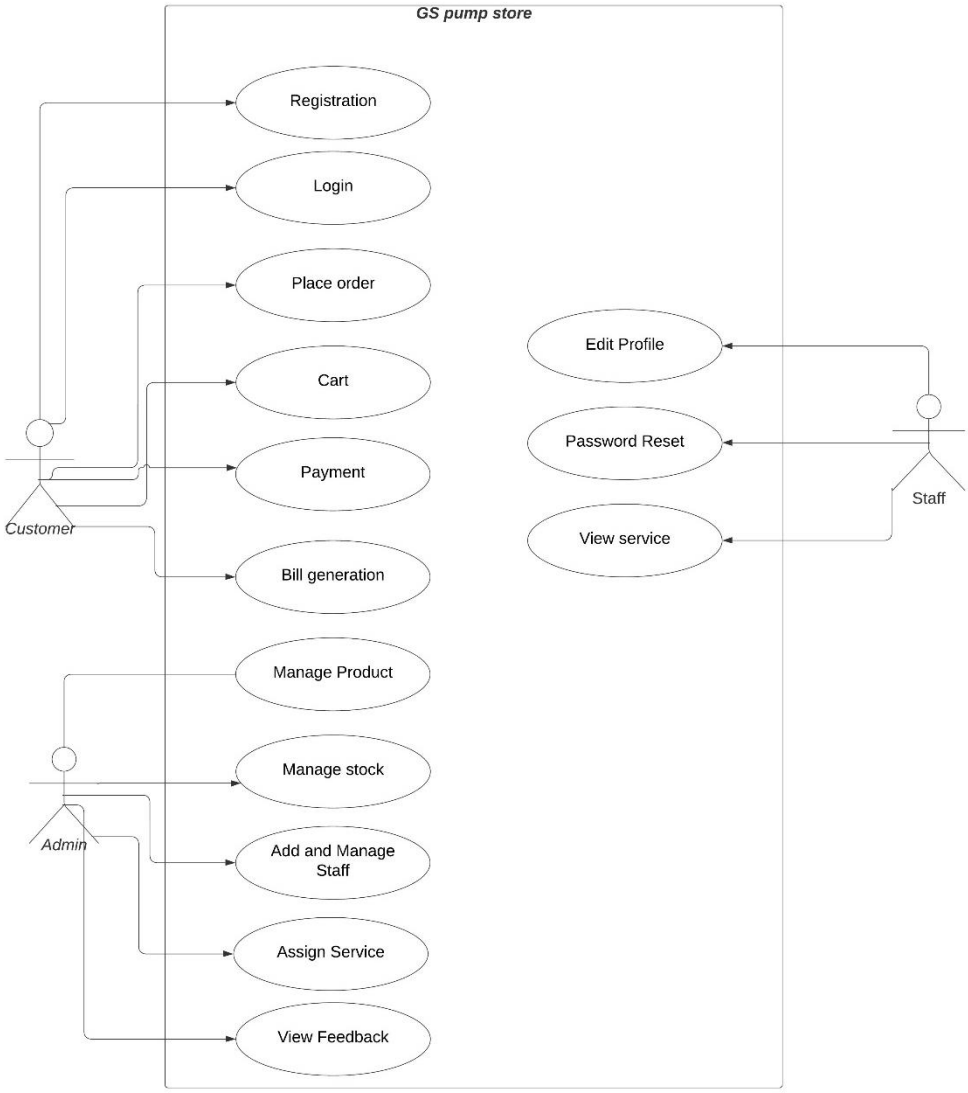
System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include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

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.

Use Case Diagram



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:

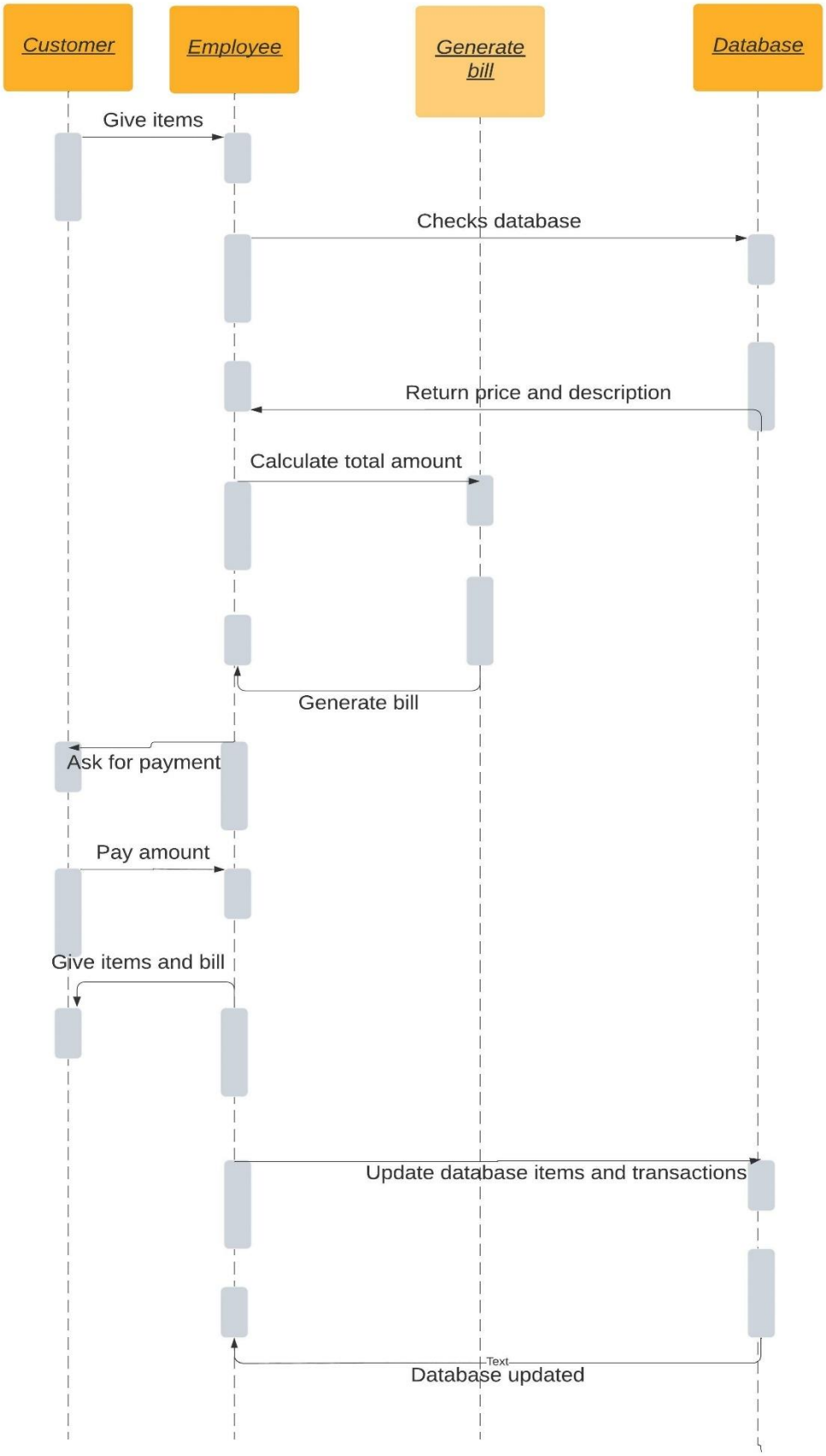
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message

- 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 sophisticated function, operation or procedure.
- 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.


Customer
Employee
Generate bill
Database
Give items
Checks database
Return price and description
Calculate total amount
Generate bill
Ask for payment
Pay amount
Give items and bill
Update database items and transactions
Text
Database updated

4.2.3 ACTIVITY DIAGRAM

The activity diagram used to describe flow of activity through a series of actions. Activity diagram is an important diagram to describe the system. The activity described as a action or operation of the system.

Initial State or Start Point

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram. For activity diagram using swimlanes, make sure the start point is placed in the top left corner of the first column.

Activity or Action State

An action state represents the non-interruptible action of objects. You can draw an action state in SmartDraw using a rectangle with rounded corners.

Action Flow

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.

Object Flow

Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.

Decisions and Branching

A diamond represents a decision with alternate paths. When an activity requires a decision prior to moving on to the next activity, add a diamond between the two activities. The outgoing alternates should be labeled with a condition or guard expression. You can also label one of the paths "else."

Guards

In UML, guards are a statement written next to a decision diamond that must be true before moving next to the next activity. These are not essential, but are useful when a specific answer, such as "Yes, three labels are printed," is needed before moving forward.

Synchronization

A fork node is used to split a single incoming flow into multiple concurrent flows. It is represented as a straight, slightly thicker line in an activity diagram. A join node joins multiple concurrent flows back into a single outgoing flow. A fork and join mode used together are often referred to as synchronization.

Time Event

This refers to an event that stops the flow for a time; an hourglass depicts it.

Merge Event

A merge event brings together multiple flows that are not concurrent.

Sent and Received Signals

Signals represent how activities can be modified from outside the system. They usually appear in pairs of sent and received signals, because the state can't change until a response is received, much like synchronous messages in a sequence diagram. For example, an authorization of payment is needed before an order can be completed.

Interrupting Edge

An event, such as a cancellation, that interrupts the flow denoted with a lightning bolt.

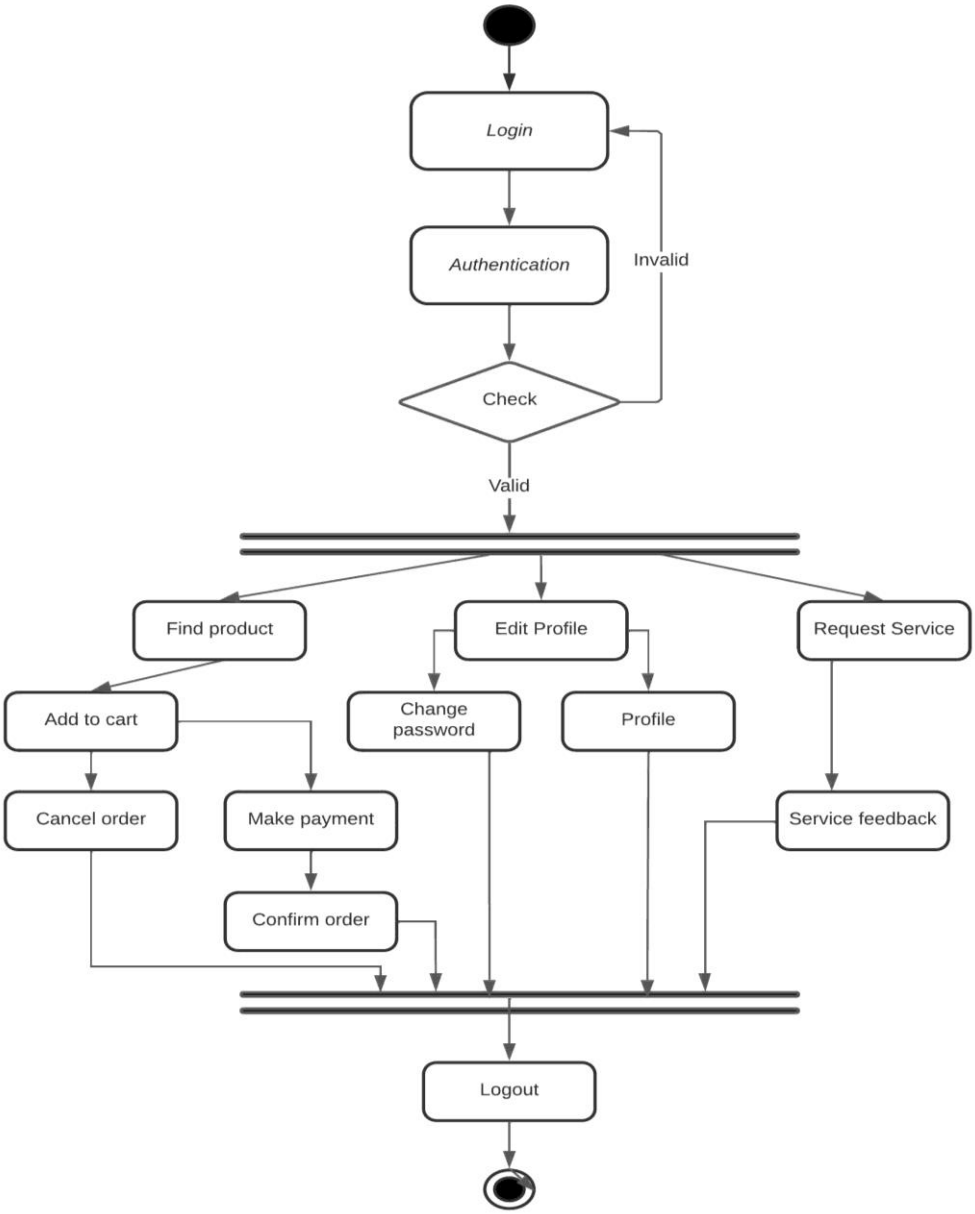
Swim lanes

Swim lanes group related activities into one column.

Final State or End Point

An arrow pointing to a filled circle nested inside another circle represents the final action state.

Activity Diagram



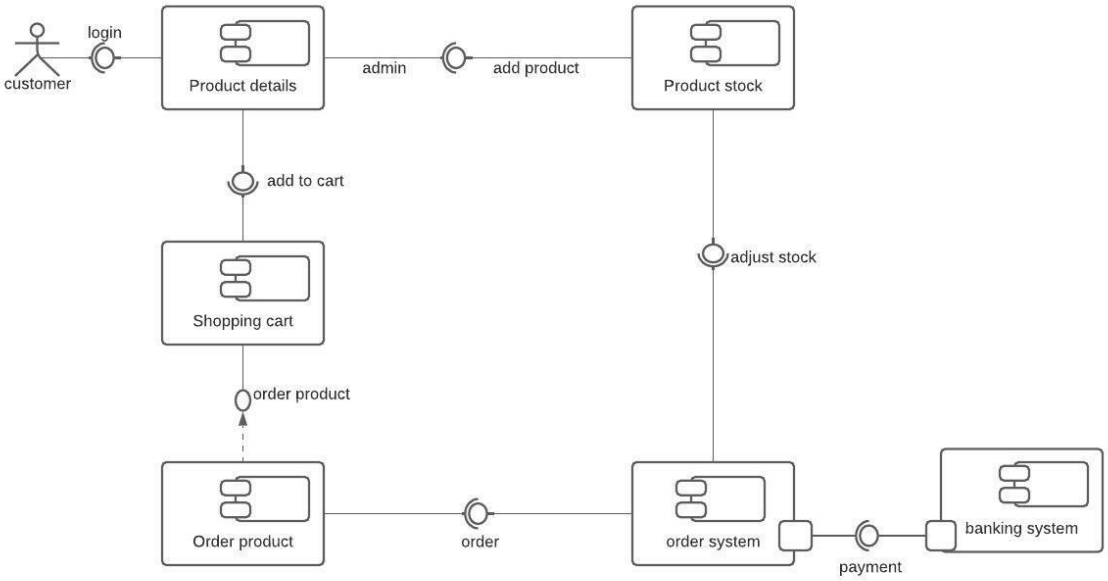
4.2.4 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems. 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. Static implementation represents the organization of the components at a particular moment.

Component Diagram



4.2.5 CLASS DIAGRAM

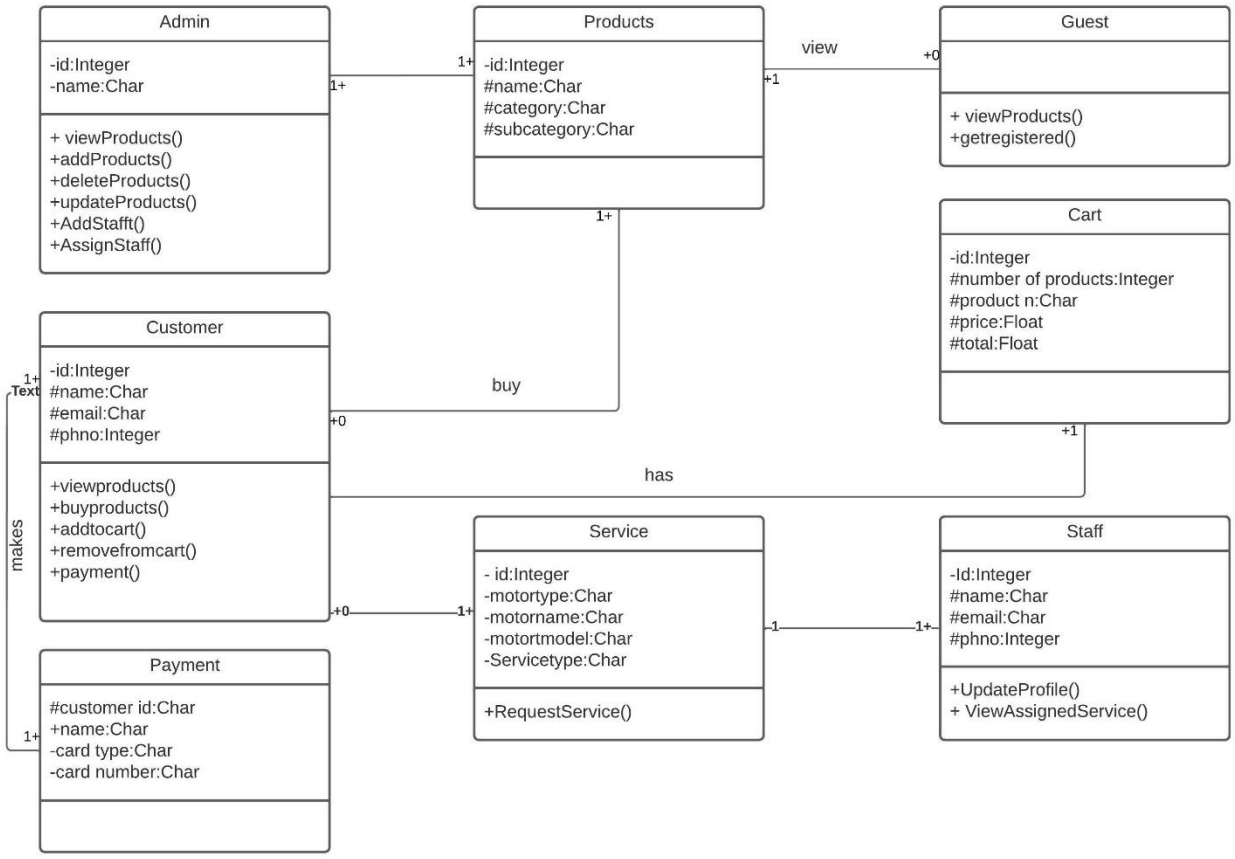
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

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 objectoriented 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.

- This is the only UML that can appropriately depict various aspects of the OOPs concept.
- Proper design and analysis of applications can be faster and efficient.
- It is the base for deployment and component diagram.

Class Diagram

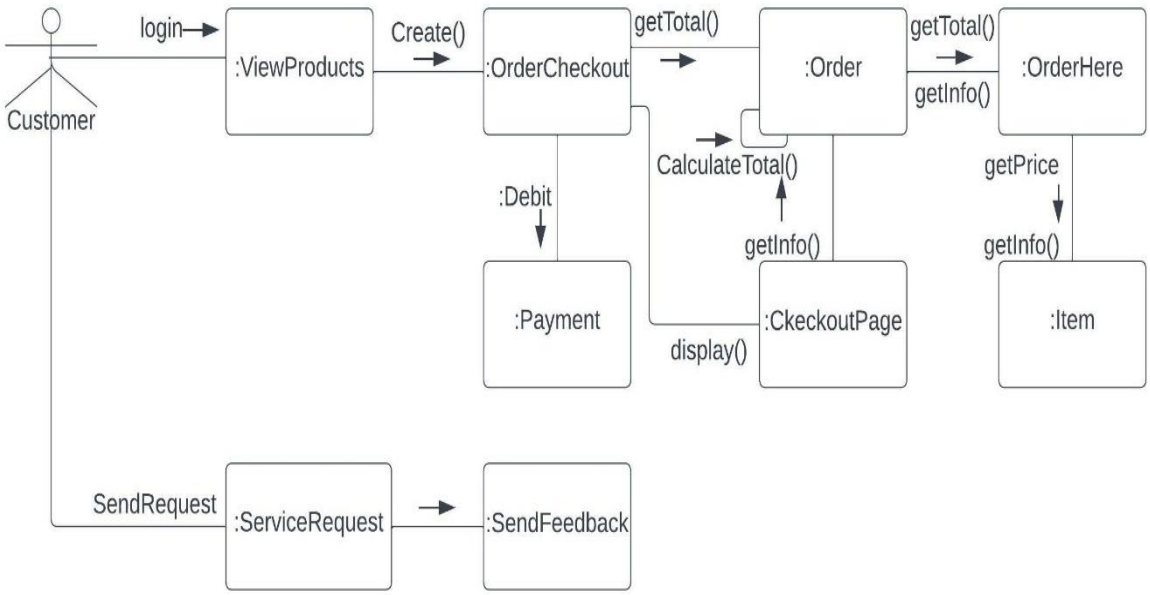


4.2.6 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. The four major components of a collaboration diagram are:

- Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.
- Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.
- Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.
- messages- Messages between objects are shown as a labeled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.


Customer
login
:ViewProducts
Create()
:OrderCheckout
getTotal()
:Order
getTotal()
getInfo()
:OrderHere
CalculateTotal()
getInfo()
getPrice
getInfo()
:Debit
:Payment
:CkeckoutPage
display()
:Item
SendRequest
:ServiceRequest
:SendFeedback

4.2.7 DEPLOYMENT DIAGRAM

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. 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.

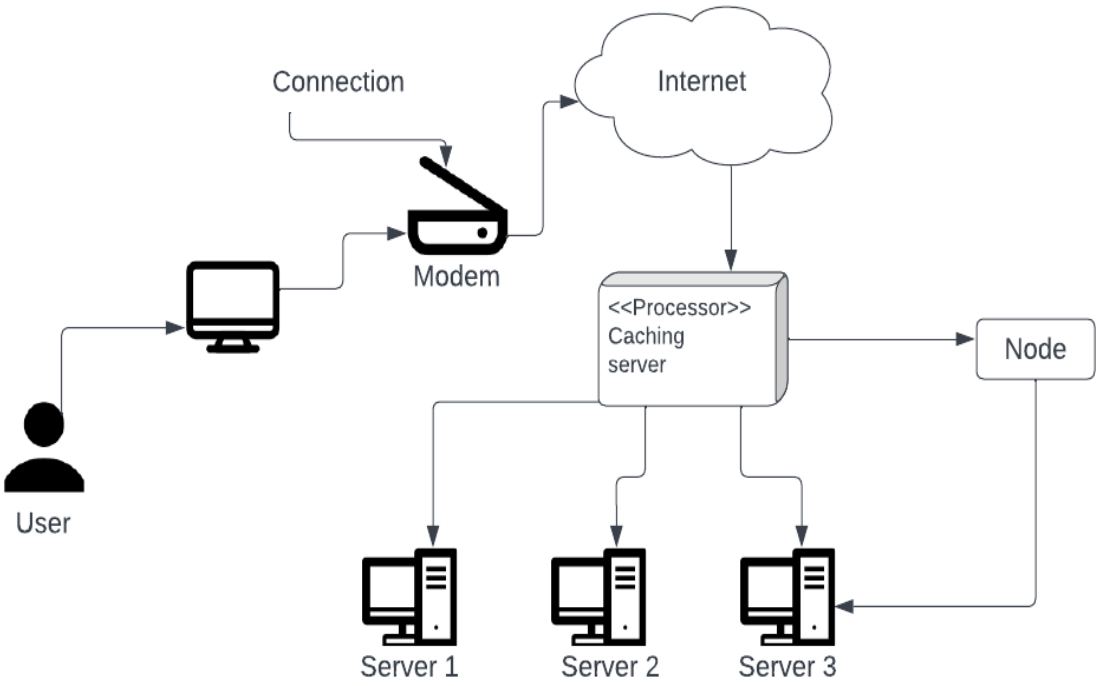
The purpose of deployment diagrams can be described as −

- 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

Deployment Diagram



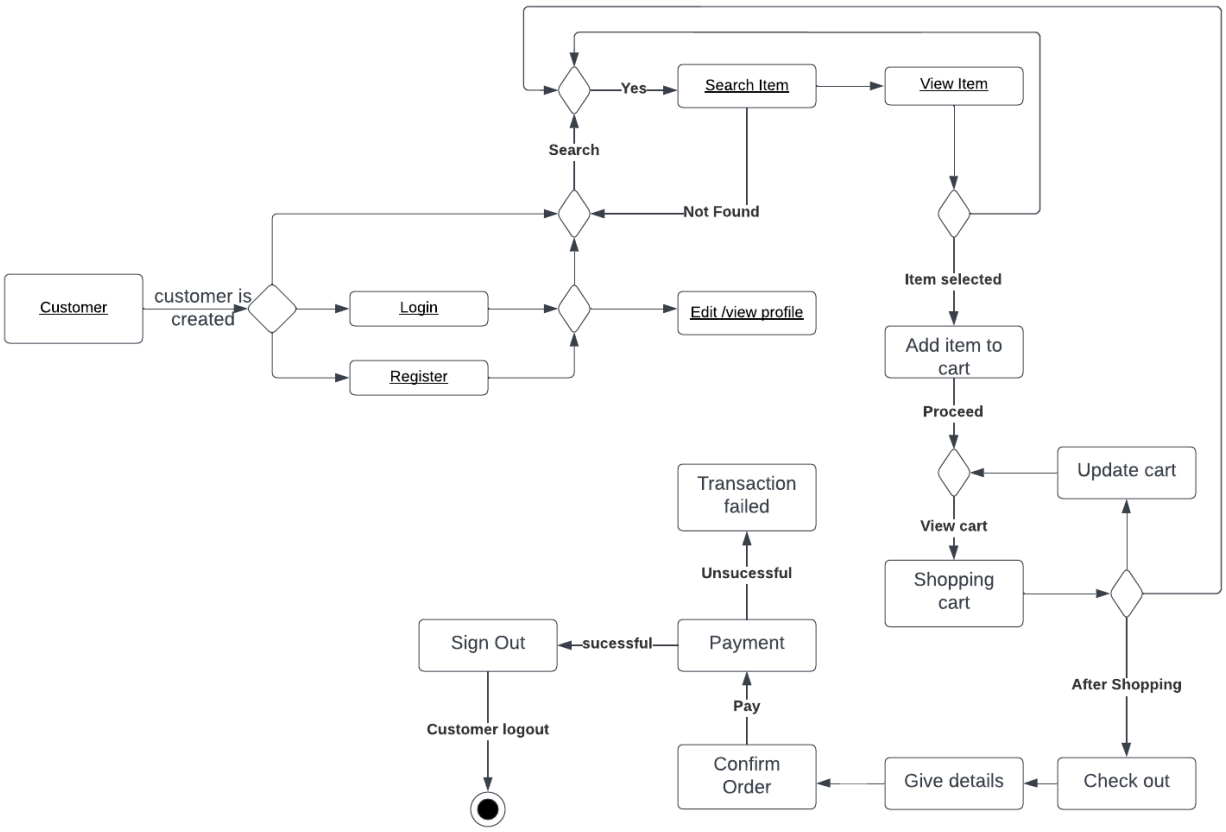
4.2.8 STATE DIAGRAM

A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. We can say that each and every class has a state but we don't model every class using State diagrams. We prefer to model the states with three or more states.

Uses of state chart diagram –

- We use it to state the events responsible for change in state (we do not show what processes cause those events).
- We use it to model the dynamic behavior of the system.
- To understand the reaction of objects/classes to internal or external stimuli.

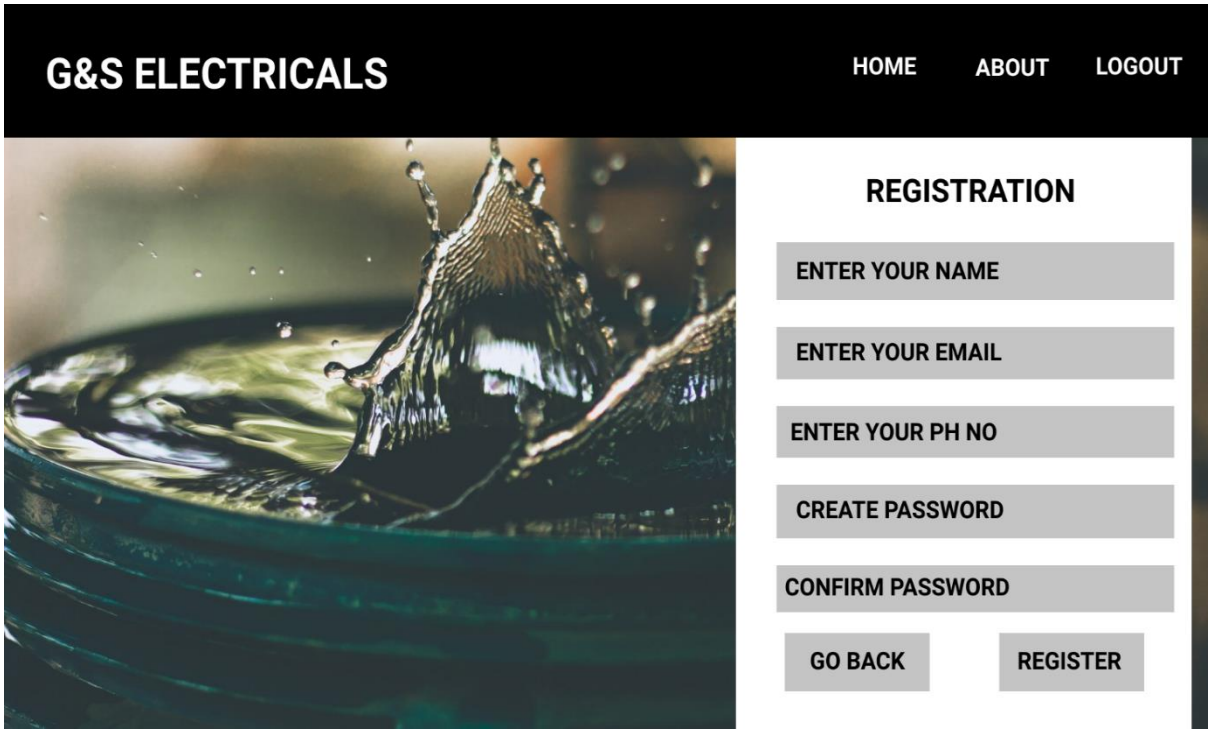
State Diagram



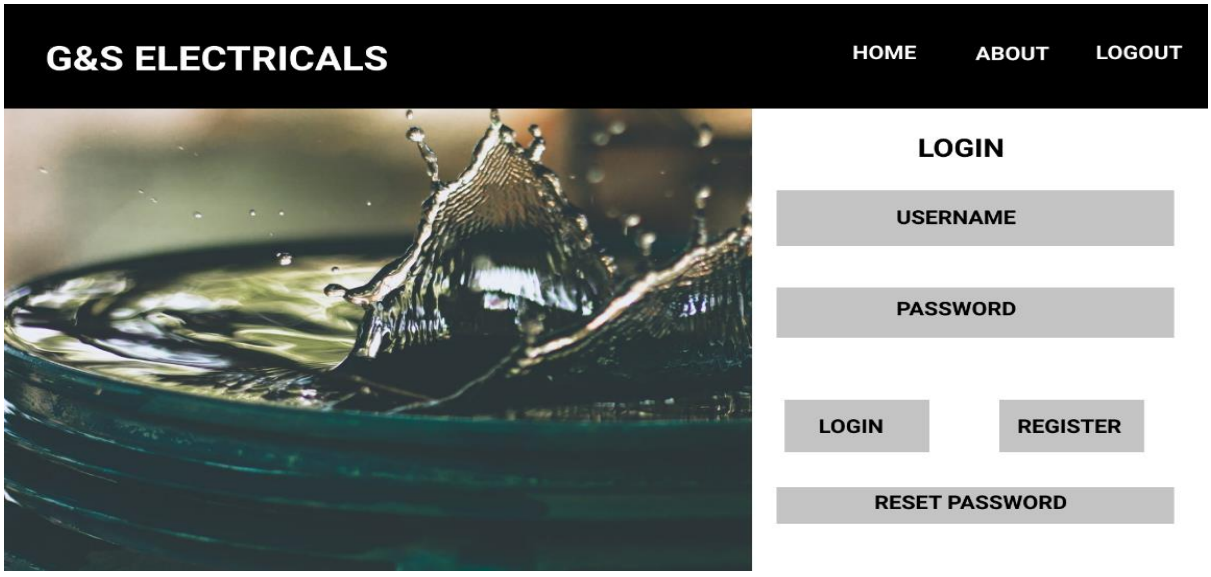
4.5 USER INTERFACE DESIGN USING FIGMA

4.5.1 INPUT DESIGN

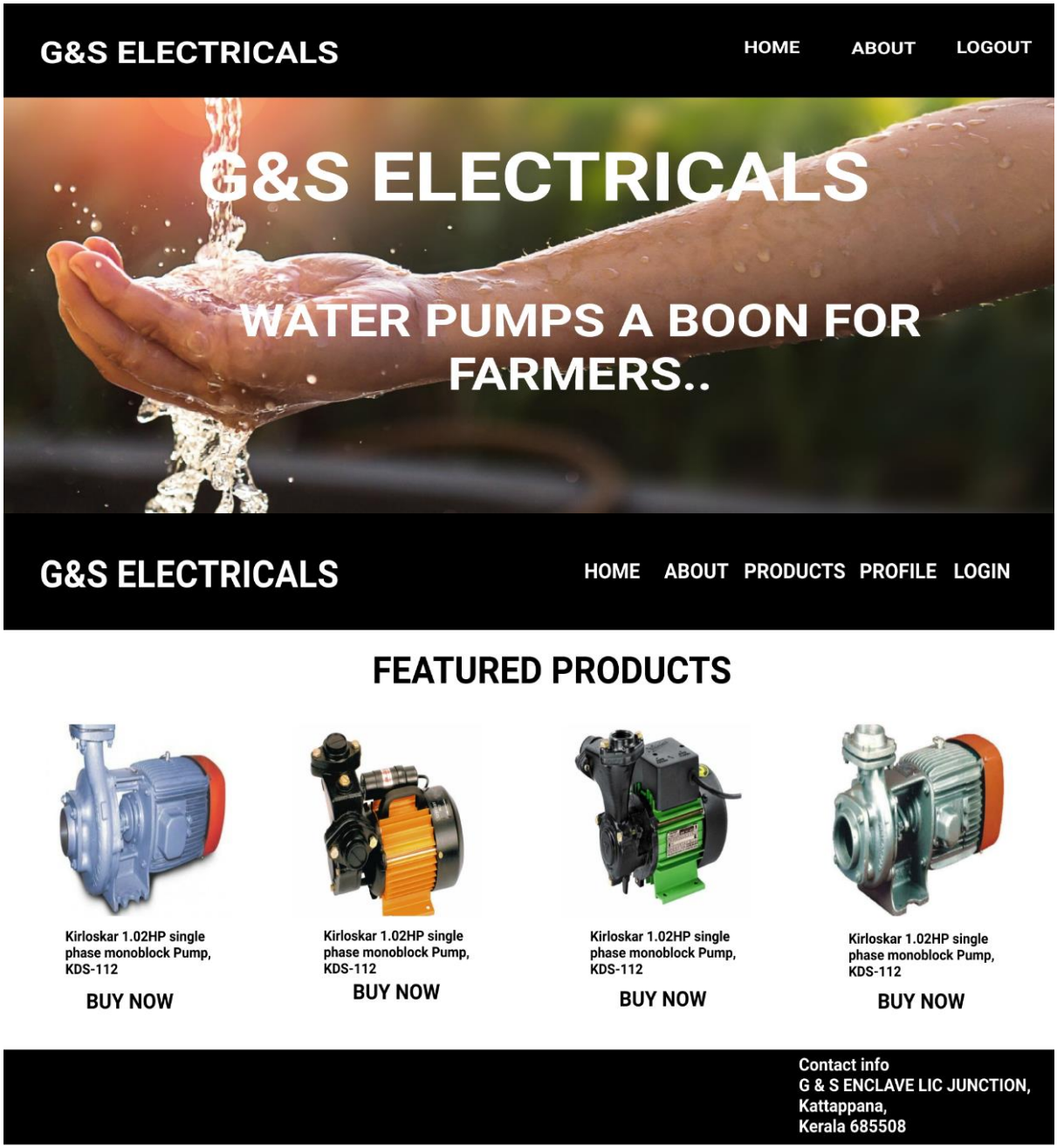
Form Name : User Registration



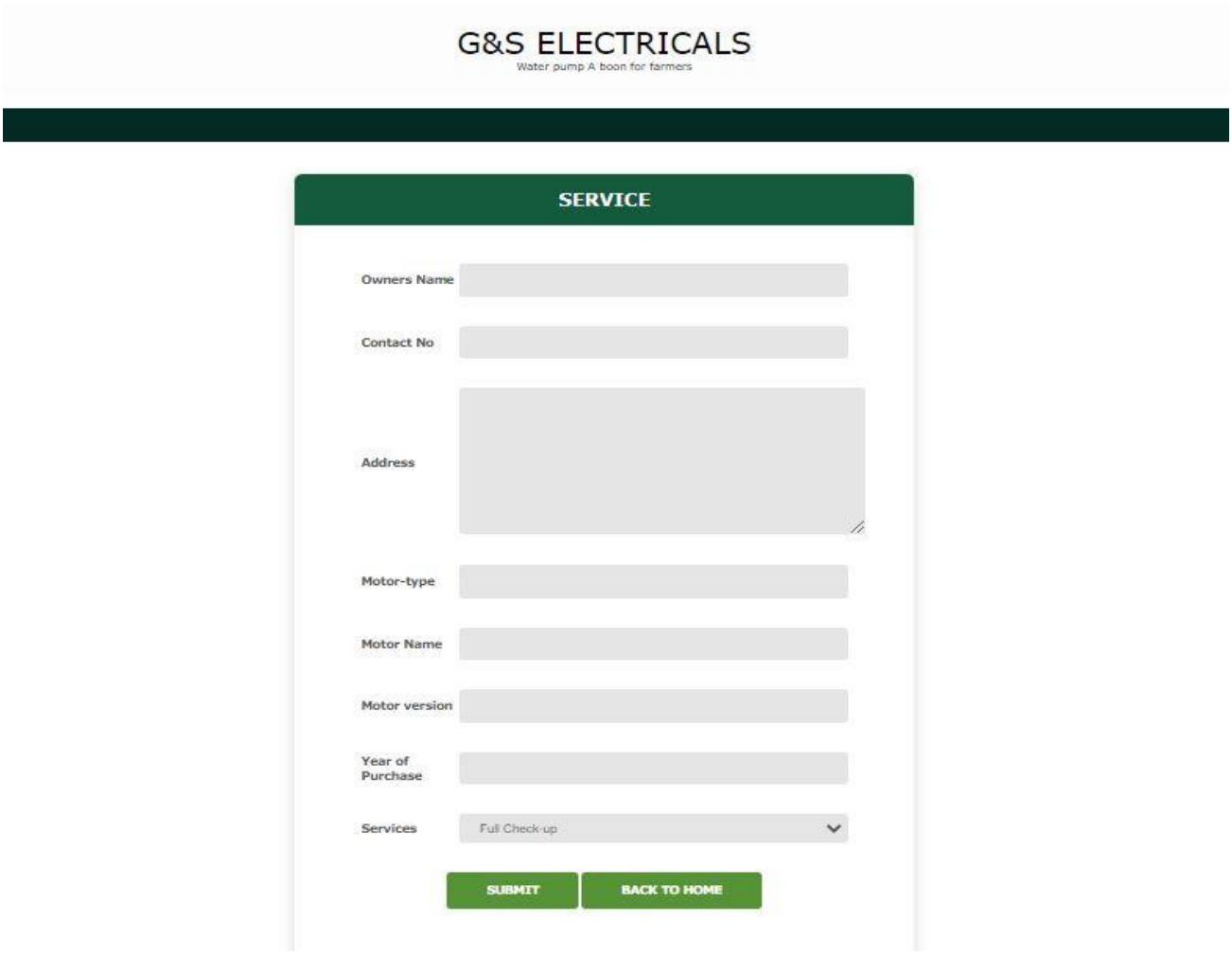
Form Name : User Login



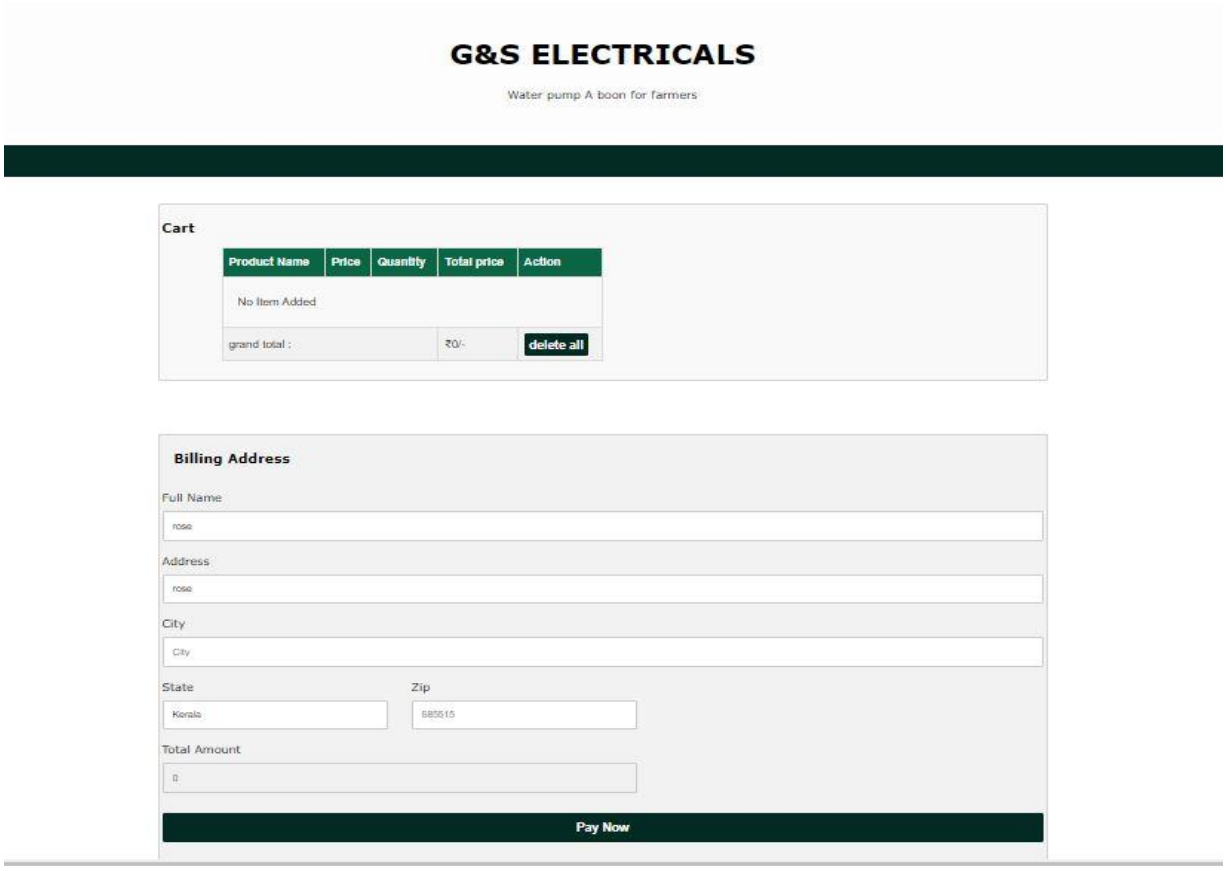
Form Name : Index page



Form Name : Service Request

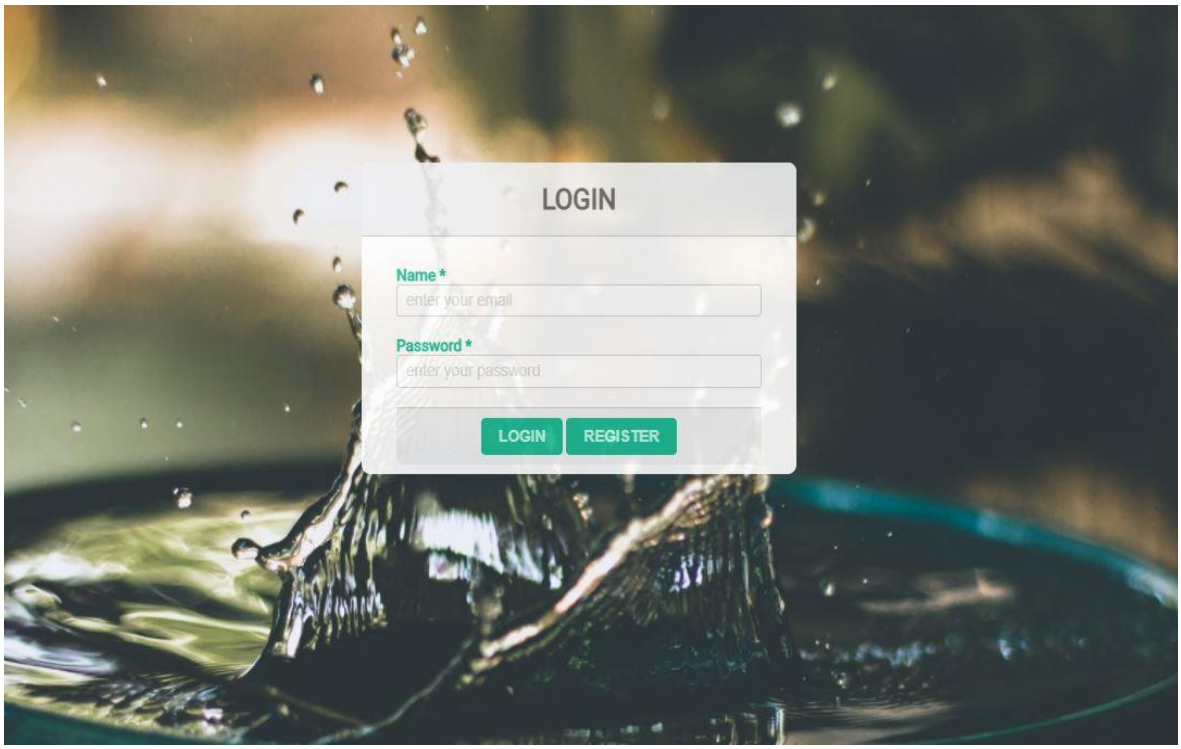


Form Name : Checkout Form

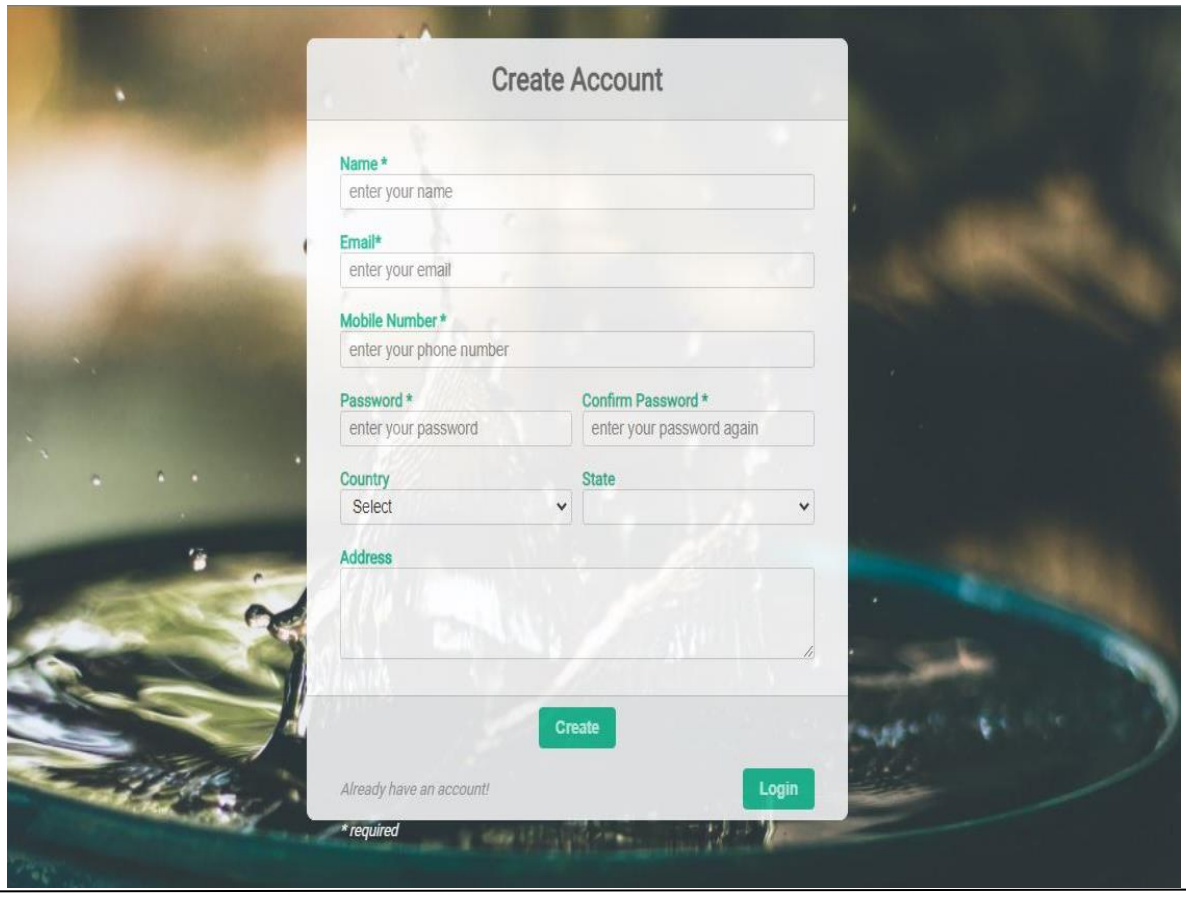


4.5.2 OUTPUT DESIGN

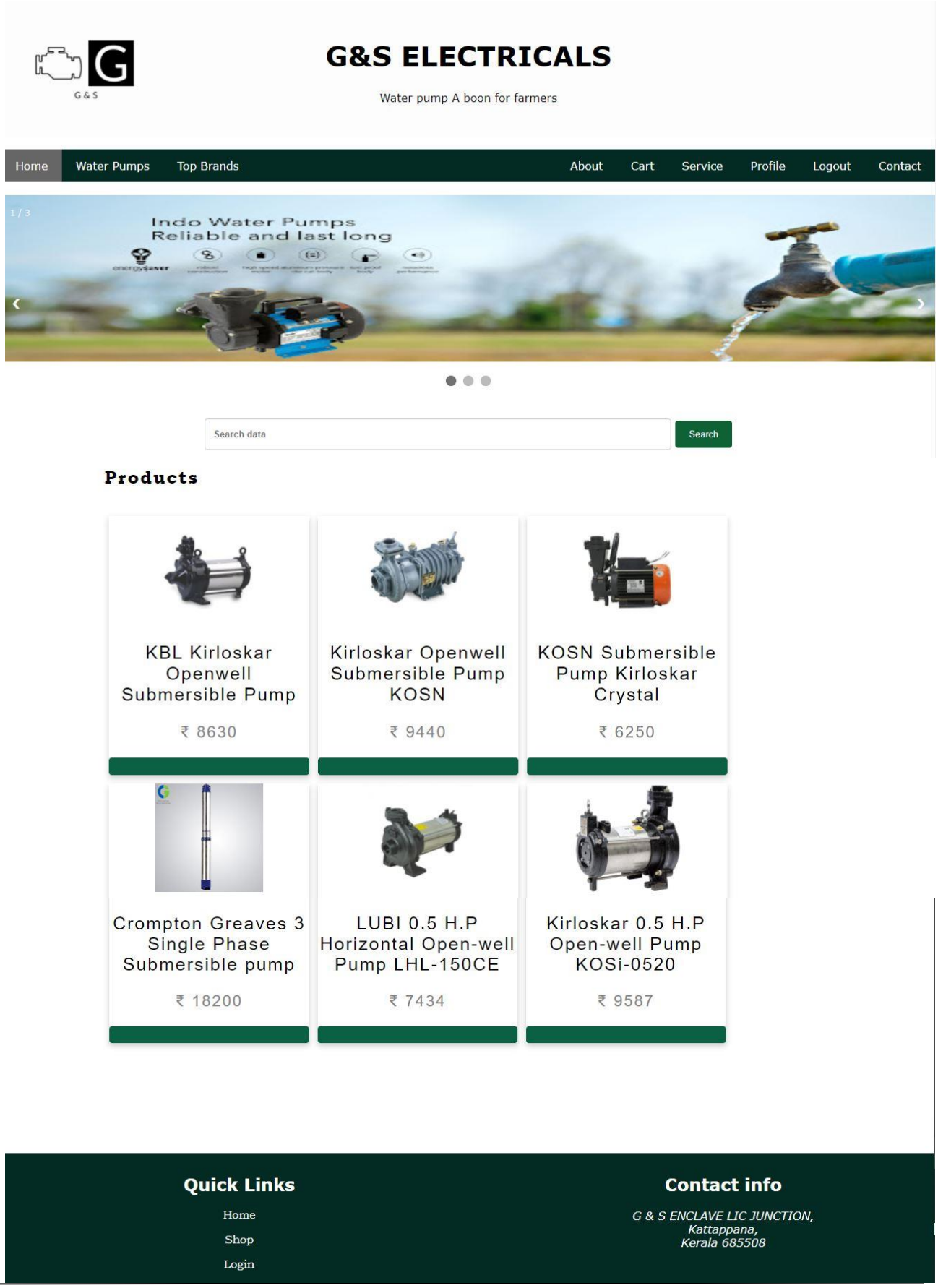
User and Admin Login



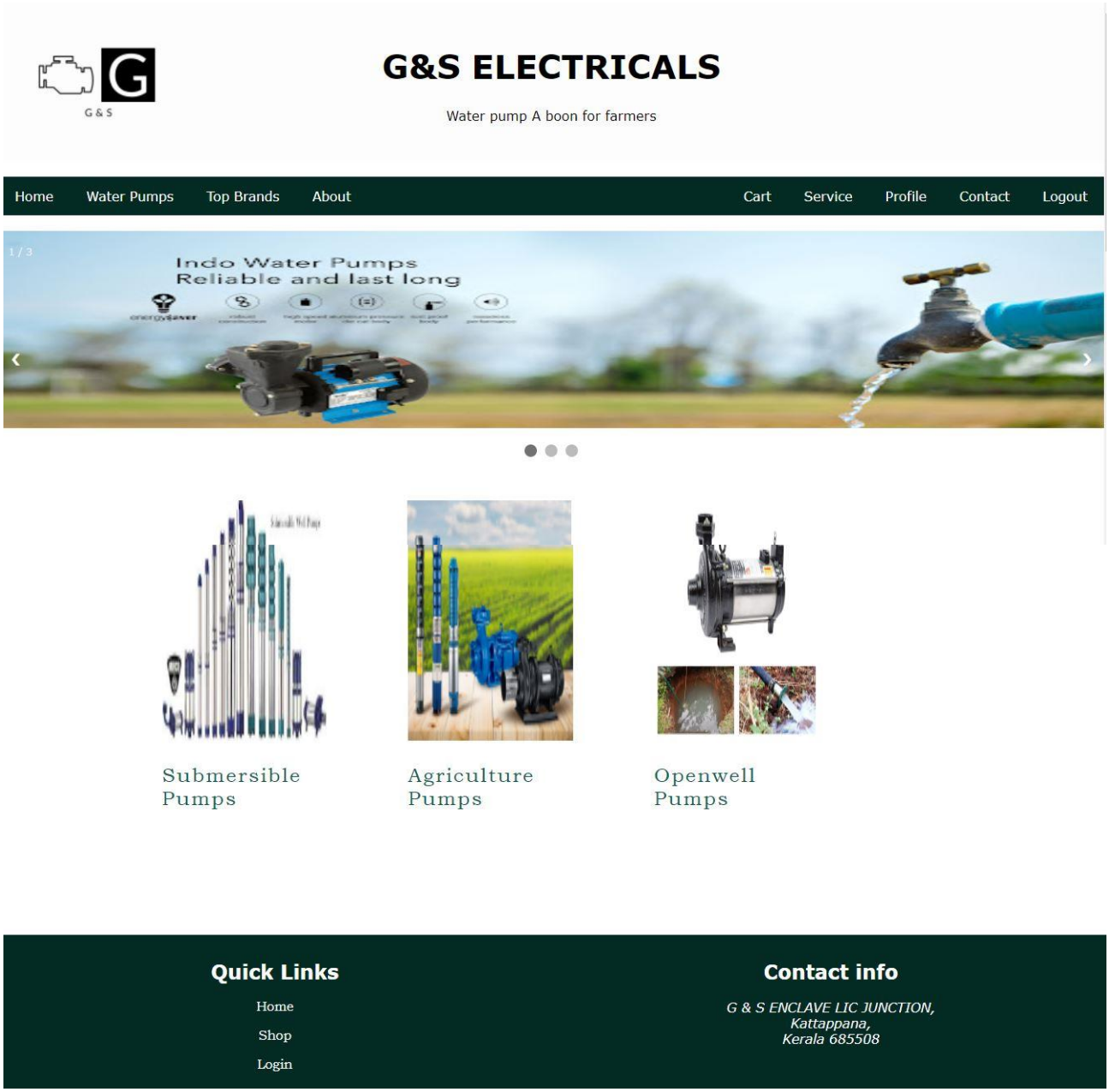
User Registration



Home Page


G&S ELECTRICALS
Water pump A boon for farmers
Home
Water Pumps
Top Brands
About
Cart
Service
Profile
Logout
Contact
Indo Water Pumps
Reliable and last long
Search data
Search
Products
KBL Kirloskar Openwell Submersible Pump
₹ 8630
Kirloskar Openwell Submersible Pump KOSN
₹ 9440
KOSN Submersible Pump Kirloskar Crystal
₹ 6250
Crompton Greaves 3 Single Phase Submersible pump
₹ 18200
LUBI 0.5 H.P Horizontal Open-well Pump LHL-150CE
₹ 7434
Kirloskar 0.5 H.P Open-well Pump KOSi-0520
₹ 9587
Quick Links
Home
Shop
Login
Contact info
G & S ENCLAVE LIC JUNCTION,
Kattappana,
Kerala 685508

Category Page



4.6 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

4.6.1 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 key are Super Key and Candidate Keys.

4.6.2 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 - rid

Field Name	Type	Size	Description
rid	int	10	Primary key of register table
name	varchar	30	Name of the customer
email	varchar	50	email id of the customer
phone	int	12	Phone number of the customer
country	varchar	30	Country of user
state	varchar	30	State of user
address	varchar	100	Address of user
password	varchar	100	password
status	int	10	To active and deactivate user

Table No 02 - tbl_usertype
Primary Key - type_id

Field Name	Type	Size	Description
type_id	int	30	Primary key of usertype table
type_name	varchar	50	Type of user(admin, user, staff)

Table No 03 - tbl_login
Primary Key - login_id
Foreign Key - rid references Register Table

Field Name	Type	Size	Description
login_id	int	30	Primary key of login table
rid	int	20	Foreign key from register table
type_id	int	30	User type

Table No 04 - tbl_prdttype
Primary Key - prd_id

Field Name	Type	Size	Description
prd_id	Int	10	Primary key of main category table
prd_name	varchar	30	Main-Category name
prd_status	int	5	To active and deactivate main-category

Table No 05 - tbl_category
Primary Key - cat_id
Foreign Key - prd_id References Product table

Field Name	Type	Size	Description
cat_id	Int	10	Primary key of category table
prd_id	Int	10	Foreign key main category table
cat_img	varchar	255	Category image
cat_name	varchar	30	Category name
cat_status	int	5	To active and deactivate category

Table No 06 - tbl_subcategory
Primary Key - sub_id
Foreign Key - cat_id References Category table

Field Name	Type	Size	Description
sub_id	int	20	Primary key of subcategory table
cat_id	int	20	Foreign key of category table
sub_name	varchar	255	product name
sub_desc	varchar	255	Description of the product
sub_spec	varchar	255	Specification of the product
sub_img	varchar	255	Image of the product
sub_price	varchar	20	Price of the product
sub_status	int	3	To active and deactivate sub-category

Table No 07 - tbl_checkout
Primary Key - ch_id
Foreign Key - rid references Register Table

Field Name	Type	Size	Description
ch_id	Int	10	Primary key of checkout table
rid	Int	10	Foreign key from register table
city	varchar	100	City of customer
zip	int	15	Pincode of customer
ctype	varchar	50	type of the card
cnum	varchar	30	card number
cname	varchar	50	name on the card used
cvv	int	5	The three digit CVV
exp	varchar	20	Card expiry date

Table No 08 - tbl_cart
Primary Key - cart_id
Foreign Key - rid references Register Table

Field Name	Type	Size	Description
cart_id	Int	10	Primary key of payment table
rid	Int	10	Foreign key from register table
pname	varchar	50	name of the product
pprice	varchar	100	price of product
qty	int	20	quantity of the product

Table No 09 - tbl_payment
Primary Key - pid

Field Name	Type	Size	Description
pid	Int	10	Primary key of payment table
pname	varchar	50	Payee name
amount	int	11	Total amount
payment_status	varchar	50	Pending or complete
payment_id	varchar	50	id of that particular payment
added_on	datetime	30	Date

Table No 10 - tbl_service
Primary Key - ser_id

Field Name	Type	Size	Description
ser_id	Int	10	Primary key of service table
oname	varchar	50	Owner's name
ophn	varchar	50	Owner's contact number
oaddress	varchar	100	Owner's address
mtype	varchar	50	Motor type
mname	varchar	50	Name of the motor
mno	varchar	30	Motor model number
pyear	varchar	100	year of purchase
service	varchar	100	Motor services
status	int	20	Yes or no
assign	varchar	50	Assigned Staff

Table No 11 - tbl_staff
Primary Key - staff_id

Field Name	Type	Size	Description
staff_id	Int	20	Primary key of staff table
staff_gender	varchar	50	Gender of staff
staff_dob	varchar	100	Date of birth of staff
staff_spec	varchar	150	Details of staff
staff_quali	varchar	250	Qualifications of staff
staff_idp	varchar	300	Id proof of staff
staff_photo	varchar	300	Photo of staff

Table No 12 - tbl_assign
Primary Key - assign_id
Foreign Key - staff_id, ser_id References Staff and Service Table

Field Name	Type	Size	Description
assign_id	Int	20	Primary key of assign table
staff_id	Int	20	Foreign key from staff table
ser_id	Int	20	Foreign key from service table

Table No 13 - tbl_service_feedback
Primary Key - cf_id
Foreign Key - rid references Register Table

Field Name	Type	Size	Description
cf_id	Int	20	Primary key of feedback table
rid	Int	20	Foreign key from register table
cfsub	varchar	100	Message subject
cfmsg	varchar	100	Message content
cftime	timestamp	6	Current date and time

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 its 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 where 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. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

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.

5.2.5 Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strictrequirements. 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 productdevelopment.

Benefits of Automation Testing,

- ✓ Detailed reporting capabilities - Automation testing uses well-crafted test cases for variousscenarios. These scripted sequences can be incredibly in-depth, and provide detailed reportsthat 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 analyzea wider test coverage than humans may be able to.

5.2.6 Selenium Testing

Selenium is an open-source and a portable automated software testing tool for testing web applications. It has capabilities to operate across different browsers and operating systems. Selenium is not just a single tool but a set of tools that helps testers to automate web-based applications more efficiently. You can use multiple programming languages like Java, C#, Python etc. to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

Test cases for a Login Page

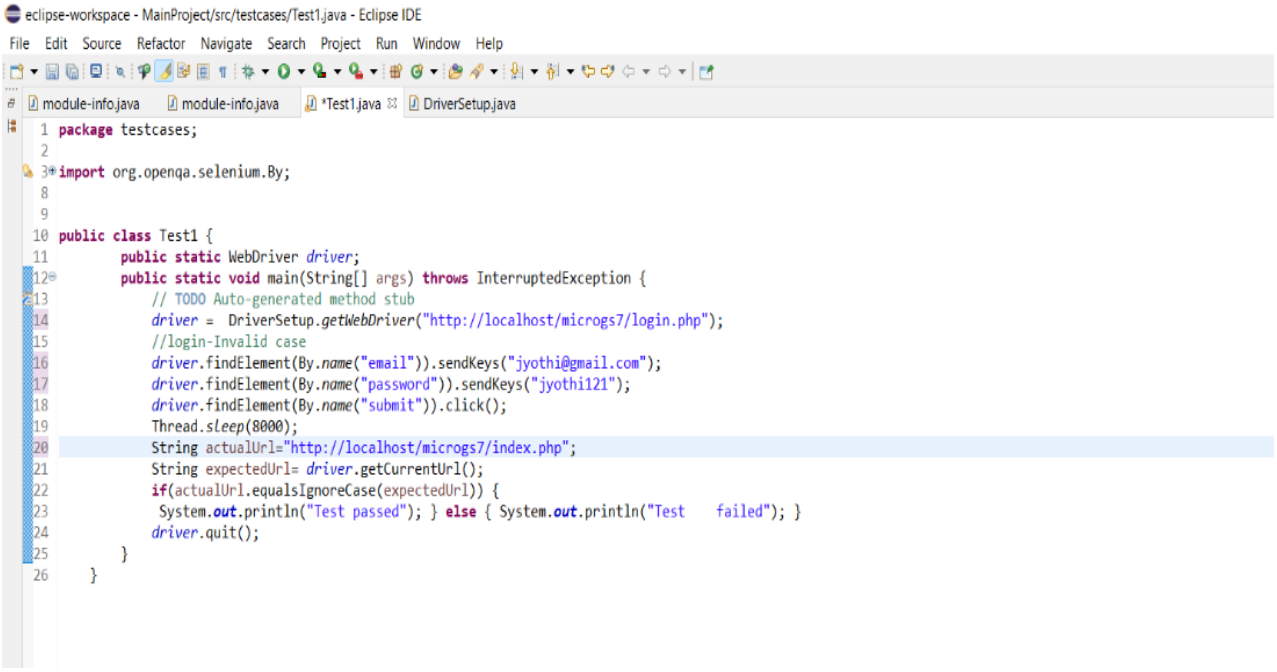
Project Name: G&S Motor Pump Store					
Login Test Case					
Test Case ID: Fun_1			**Test Designed By: Jyothika Suresh**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 21-05-2022		
Module Name: Login Screen **Test Title:** Verify login with validemail and password			**Description:** 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 toLogin Page		Login Page should be displayed	Login page displayed	Pass
2	Provide Valid Email Id	User Name: jyothi@gmail.com	User should be able to Login	User Logged inand navigated to Subadmin Dashboard with records	Pass
3	Provide Valid Password	Password: jyothi121			
4	Click on Sign In button				
5	Provide Invalid Email Id or password	Email Id: user.gmail. 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				

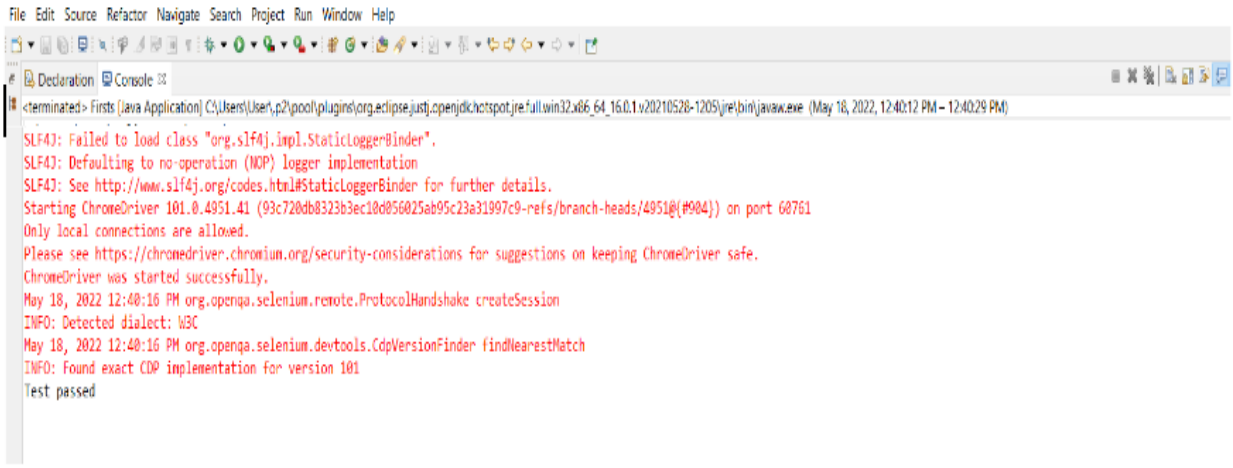
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database

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/microgs7/login.php");
            //login-Invalid case
            driver.findElement(By.name("email")).sendKeys("jyothi@gmail.com");
            driver.findElement(By.name("psw")).sendKeys("jyothi121");
            driver.findElement(By.name("submit")).click();
            Thread.sleep(8000);
            String actualUrl="http://localhost/microgs7/index.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
1 package testcases;
2
3 import org.openqa.selenium.By;
8
9
10 public class Test1 {
11 public static WebDriver driver;
12 public static void main(String[] args) throws InterruptedException {
13 // TODO Auto-generated method stub
14 driver = DriverSetup.getWebDriver("http://localhost/microgs7/login.php");
15 //login-Invalid case
16 driver.findElement(By.name("email")).sendKeys("jyothi@gmail.com");
17 driver.findElement(By.name("password")).sendKeys("jyothi121");
18 driver.findElement(By.name("submit")).click();
19 Thread.sleep(8000);
20 String actualUrl="http://localhost/microgs7/index.php";
21 String expectedUrl= driver.getCurrentUrl();
22 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
23 System.out.println("Test passed"); } else { System.out.println("Test failed"); }
24 driver.quit();
25 }
26 }


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 PM)
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

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.

6.2.1 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.

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 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

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".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The project entitled G&S Motor-Pump Store has been developed with much care 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 motor pumps from a shop.

In this project, the user is provided with an e-commerce web site that can be used to buy agricultural motor pump online. It provides the user with a catalog of different product available for purchase in the store. In order to facilitate online purchase a shopping cart is provided to the user. The current system introduces bill generation, and also contains a workshop. This system provides lots of advantages like search products, view profile of products, enhanced user interface, payment options, and many more.

7.2FUTURE SCOPE

- Customers can able to do advanced search options.
- Data security can be enhanced.
- Transferring data with high security.
- Communication through chats.
- Newsletter subscription can be added.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, "*System Analysis and Design*", 2009.
- Roger S Pressman, "*Software Engineering*", 1994.
- PankajJalote, "So*ftware engineering*: a precise approach", 2006.
- James lee and Brent ware Addison, "Open source web development with LAMP", 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1Sample Code

Addstaff.php

```
<?php
  include 'dbcon.php';
  session_start();
  if (isset($_POST["btn"])){
  $name=$_POST["sname"];
  //echo $name;
  $email=$_POST["semail"];
  //echo $email;
  $mobile=$_POST["sphn"];
  //echo $mobile;
  $pass=md5($_POST["spass"]);
  //echo $pass;
  $cnt=$_POST["country"];
  //echo $cnt;
  $state=$_POST["state"];
  //echo $state;
  $address=$_POST["saddress"];
  //echo $address;

  $sql="INSERT INTO `reg`(`name`, `email`, `phone`, `country`, `state`, `address`, `pass`, `status`)
  VALUES ('$name','$email','$mobile','$cnt','$state','$address','$pass',0)";
  //header("location:view.php");

  if(mysqli_query($con,$sql))
  {
  $sql2= "SELECT * FROM `reg` WHERE `email`='$email'";
  $data=mysqli_query($con,$sql2);
  if($res=mysqli_fetch_assoc($data))
  {
  $sql3="SELECT * FROM `usertype` WHERE `type_name`='staff'";
  $data1=mysqli_query($con,$sql3);
  if($row=mysqli_fetch_assoc($data1))
  {
  $reg=$res['rid'];
  $type=$row['type_id'];
  $sql4="INSERT INTO `login` (`rid`,`type_id`) VALUES ('$reg','$type')";
  if(mysqli_query($con,$sql4))
  {
  echo "success";
  }
  else
  {
  echo mysqli_errno($con);
  }
  }}
  }}
?>
```

```
<!DOCTYPE html>

<head>
	<meta charset="UTF-8">
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
	<script type= "text/javascript" src = "countries.js"></script>
	<link rel="stylesheet" href="css/admin.css">

	<title>GS Admin</title>
	<style>
	input[type=text], select, textarea {
	width: 100%;
	padding: 12px;
	border: 1px solid #ccc;
	border-radius: 4px;
	box-sizing: border-box;
	margin-top: 6px;
	margin-bottom: 16px;
	resize: vertical;
	}

	input[type=submit] {
	background-color: #0e6235;
	color: white;
	padding: 12px 20px;
	border: none;
	border-radius: 4px;
	cursor: pointer;
	}
	input[type=button] {
	background-color: #0e6235;
	color: white;
	padding: 12px 20px;
	border: none;
	border-radius: 4px;
	cursor: pointer;
	}
	input[type=submit]:hover {
	background-color: #2a965c;
	}

	.container {
	border-radius: 5px;
	background-color: #f2f2f2;
	padding: 20px;
	margin-left:2%;
	margin-top:3%;
	}
	</style>

</head>
<body>

	<!-- SIDEBAR -->
	<div class="sidenav">
```

```
<a class="navbar-brand" href="/">
<div class="logo-image">
<img src="image/logo3.png" class="img-fluid"/>
</div>
</a>
<a href="aregdtls.php">Registration Details</a>
<a href="addmcat.php">Category Type</a>
<a href="addcat.php">Category</a>
<a href="addprd.php">Add Product</a>
<a href="amsgs.php">Messages</a>
<br>
<a href="afeed.php">Service Feedback</a>
<br>
<a href="addstaff.php">Add Staff</a>
<a href="aserview.php">Assign Service</a>
<a href="asgndtls.php">View Assigned Services</a>
<br><br><br><br><br>
<a href="login.php" ><b>Logout<b></a>
<!--<button class="dropdown-btn">Dropdown
<i class="fa fa-caret-down"></i>
</button>
<div class="dropdown-container">
<a href="#">Link 1</a>
<a href="#">Link 2</a>
<a href="#">Link 3</a>
</div>
<a href="#contact">Search</a> -->
</div>
<!-- SIDEBAR -->

<!-- CONTENT -->
<section id="content">


<!-- MAIN -->
<main>
<div class="head-title">
<div class="left">
<h1>Dashboard</h1>
</div>
</div>
<div class="container" >
<h3>Add Category</h3>
<form method="POST" action="#" enctype="multipart/form-data">
<br><label>Staff Name:</label>
<input type="text" id="sname" name="sname" placeholder="Add Staff name">
<br><label>Staff Email:</label>
<input type="text" id="semail" name="semail" placeholder="Add Staff Email">
<br><label>Password:</label>
                                        <input type="text" id="spass" name="spass"
    placeholder="Add Staff Password">
<br><label>Contact Number:</label>
                                        <input type="text" id="sphn" name="sphn"
    placeholder="Add Staff Contact Number">
```

```
<br><label>Country:</label>
                                        <select onchange="print_state('state',this.selectedIndex);"
        id="country" class="form-input" name ="country"></select>

<br><label>State:</label>
<select name ="state" id ="state" class="form-input"></select>
<script language="javascript">print_country("country");</script>

<br><label>Address:</label>
                                        <textarea id="saddress" name="saddress"
        placeholder="Add Staff Address"></textarea>
<input type="submit" name="btn" value="Add">
</form>
</div>

<!--Staff view-->
<div class="table-data">
<div class="order">
<div class="head">
<h3>Type Details</h3>
</div>
<table>
<thead>
<tr>
<th>Staff Name</th>
<th>Staff Email</th>
<th>Staff Phone </th>
<th>Country </th>
<th>State </th>
<th>Staff Address </th>
<th>Staff Password </th>
<th>Edit Staff</th>
</tr>
</thead>
<?php
$result=mysqli_query($con, "SELECT * FROM `reg`");
while ($res=mysqli_fetch_array($result)){
//print_r($row);
?>
<tbody>
                        <tr>
<td><?php echo $res["name"];?></td>
        <td><?php echo $res["email"];?></td>
                <td><?php echo $res["phone"];?></td>
                <td><?php echo $res["country"];?></td>
                <td><?php echo $res["state"];?></td>
                <td><?php echo $res["address"];?></td>
                <td><?php echo $res["pass"];?></td>
<td><a href="updstaff.php?edit_id=<?php echo $res["rid"];?>">
                <input type="button" value="Edit"></td>
                </tr>
                </tbody>
<?php
}
?>
```

```
        </table>
        <td><a href="createpdf.php"><input type="button" value="Generate pdf"></td>
        </div>
        </div>

        </main>
        <!-- MAIN -->
        </section>
        <!-- CONTENT -->

<script>
        var dropdown = document.getElementsByClassName("dropdown-btn");
        var i;

        for (i = 0; i < dropdown.length; i++) {
        dropdown[i].addEventListener("click", function() {
        this.classList.toggle("active");
        var dropdownContent = this.nextElementSibling;
        if (dropdownContent.style.display === "block") {
        dropdownContent.style.display = "none";
        } else {
        dropdownContent.style.display = "block";
        }
        });
        }
        </script>
</body>
</html>
```

AssignStaff.php

```
        <?php
        include 'dbcon.php';
        session_start();
        $id=$_GET['aab'];
        ?>
        <!DOCTYPE html>
        <html lang="en">
        <head>
                <meta charset="UTF-8">
                <meta name="viewport" content="width=device-width, initial-scale=1.0">
                <!-- Boxicons -->
                <link href='https://unpkg.com/boxicons@2.0.9/css/boxicons.min.css' rel='stylesheet'>
                <!-- My CSS -->
                <link rel="stylesheet" href="css/admin.css">
                <title>GS Admin</title>
                <style>
                input[type=text], select, textarea {
                width: 100%;
                padding: 12px;
                border: 1px solid #ccc;
                border-radius: 4px;
                box-sizing: border-box;
```

```
            margin-top: 6px;
            margin-bottom: 16px;
            resize: vertical;
            font-weight: bold;
            }
            input[type=submit] {
            background-color: #0e6235;
            color: white;
            padding: 12px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            }
</head>
<body>

<!-- SIDEBAR
<div class="sidenav">
<a class="navbar-brand" href="/">
<div class="logo-image">
<img src="image/logo3.png" class="img-fluid"/>
</div>
</a>

<a href="aregdtls.php">Registration Details</a>
<a href="addmcat.php">Category Type</a>
<a href="addcat.php">Category</a>
<a href="addprd.php">Add Product</a>
<a href="amsgs.php">Messages</a>
<br>
<a href="addstaff.php">Add Staff</a>
<a href="aserview.php">Assign Service</a>
<a href="asgndtls.php">View Assigned Services</a>

<br><br><br><br><br><br><br><br><br><br>
<a href="login.php" ><b>Logout<b></a>   -->
<!--<button class="dropdown-btn">Dropdown
<i class="fa fa-caret-down"></i>
</button>
<div class="dropdown-container">
<a href="#">Link 1</a>
<a href="#">Link 2</a>
<a href="#">Link 3</a>
</div>
<a href="#contact">Search</a>
</div>
<!-- SIDEBAR -->

<!-- CONTENT -->
<section id="content">
<!-- MAIN -->
<main>
<div class="head-title">
<div class="left">
<h1>Dashboard</h1>
```

```
</div>

</div>
<div class="container" >

<div class="table-data">
<div class="order">
<div class="head">
<h3>Assign Services</h3>
</div>

<?php
$res = mysqli_query($con, "SELECT * from service WHERE `ser_id`='$id'");

while ($row = mysqli_fetch_array($res)) { ?>
<table class="table table-bordered">
<thead>

<tr>
<th colspan="2" style="text-align:center;">User Details</th>
</tr>
<tr>
<th scope="col">Owner's Name</th>
<td><?php echo $row["oname"]; ?></td>
</tr>
<tr>
<th scope="col">Contact Number</th>
<td><?php echo $row["ophn"]; ?></td>
</tr>
<tr>
<th scope="col">Address</th>
<td><?php echo $row["oaddress"]; ?></td>
</tr>
<tr>
<th scope="col">Motor type</th>
<td><?php echo $row["mtype"]; ?></td>
</tr>
<tr>
<th scope="col">Motor Name</th>
<td><?php echo $row["mname"]; ?></td>
</tr>
<tr>
<th scope="col">Model Number</th>
<td><?php echo $row["mno"]; ?></td>
</tr>
<tr>
<th scope="col">Purchase Year</th>
<td><?php echo $row["pyear"]; ?></td>
</tr>

<tr>
<th scope="col">Service</th>
<td><?php echo $row["service"]; ?></td>
</tr>
```

```
<?php
}
?>
<form action="assignstaff.php" method="POST">
<tr>

<th scope="col">Assign</th>
<td><select name="staff" id="staff">
<option>Select Staff</option>
<optgroup>

<?php
$sql1 = "SELECT * FROM `login` WHERE `type_id` = 3";
$row1 = mysqli_query($con, $sql1);
while($res1 = mysqli_fetch_assoc($row1))
{

$did = $res1['rid'];
echo $did;

$sql = "SELECT * FROM `reg` WHERE `rid` = '$did'";
$row2 = mysqli_query($con, $sql);
while($res = mysqli_fetch_assoc($row2))
{
$dname = $res['name'];
?>
<option><?php echo $dname; ?></option>
<?php
}

}
?>
</optgroup>
</select></td>

</tr>
<tr>
<th scope="col"></th>
<td>
<input type="hidden" name="id" value="<?php echo $id; ?>">
<a href=""> <input class="bg-primary text-white"type="submit" value="Assign" ></a>
</td>
</tr>
</form>
</thead>
</table>
</div>
</div>
</div>
</main>
<!-- MAIN -->
</section>
<!-- CONTENT -->
```

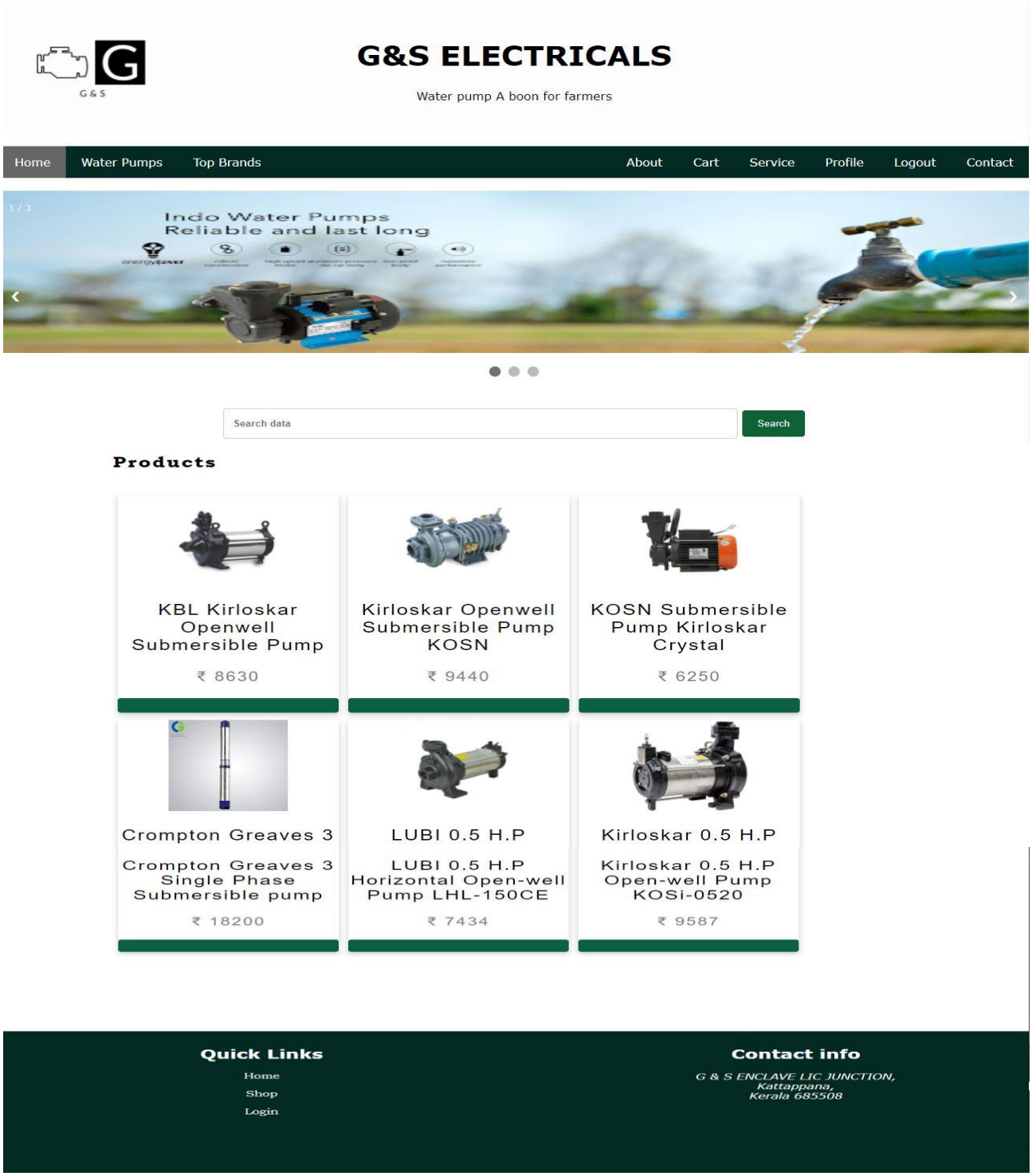
```
<script>
/* Loop through all dropdown buttons to toggle between hiding and showing its dropdown content -
This allows the user to have multiple dropdowns without any conflict */
var dropdown = document.getElementsByClassName("dropdown-btn");
var i;

for (i = 0; i < dropdown.length; i++) {
dropdown[i].addEventListener("click", function() {
this.classList.toggle("active");
var dropdownContent = this.nextElementSibling;
if (dropdownContent.style.display === "block") {
dropdownContent.style.display = "none";
} else {
dropdownContent.style.display = "block";
}
});
}
</script>
</body>
</html>
```

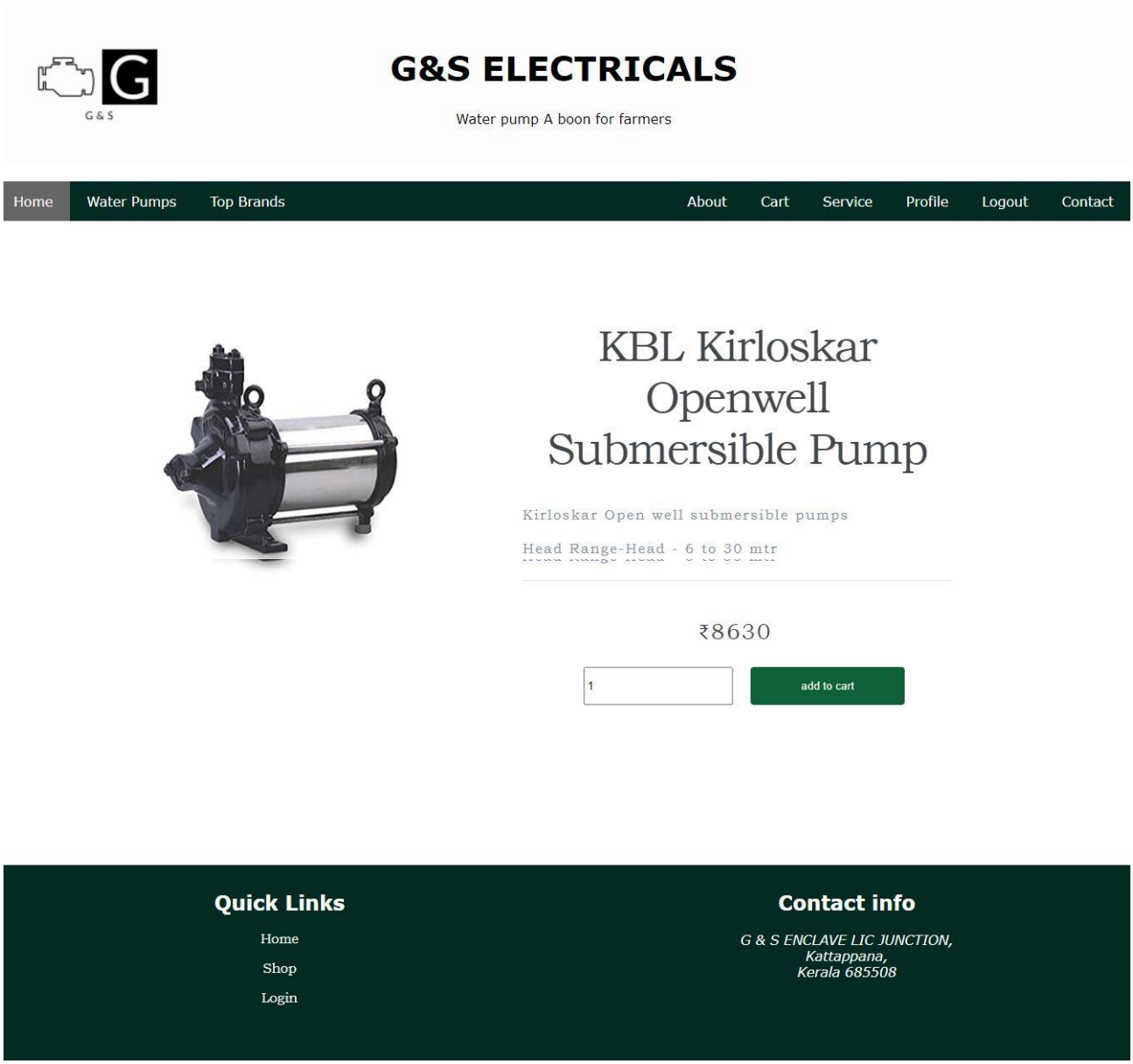
9.2 Screen Shots

CUSTOMER PAGES

Customer Home page

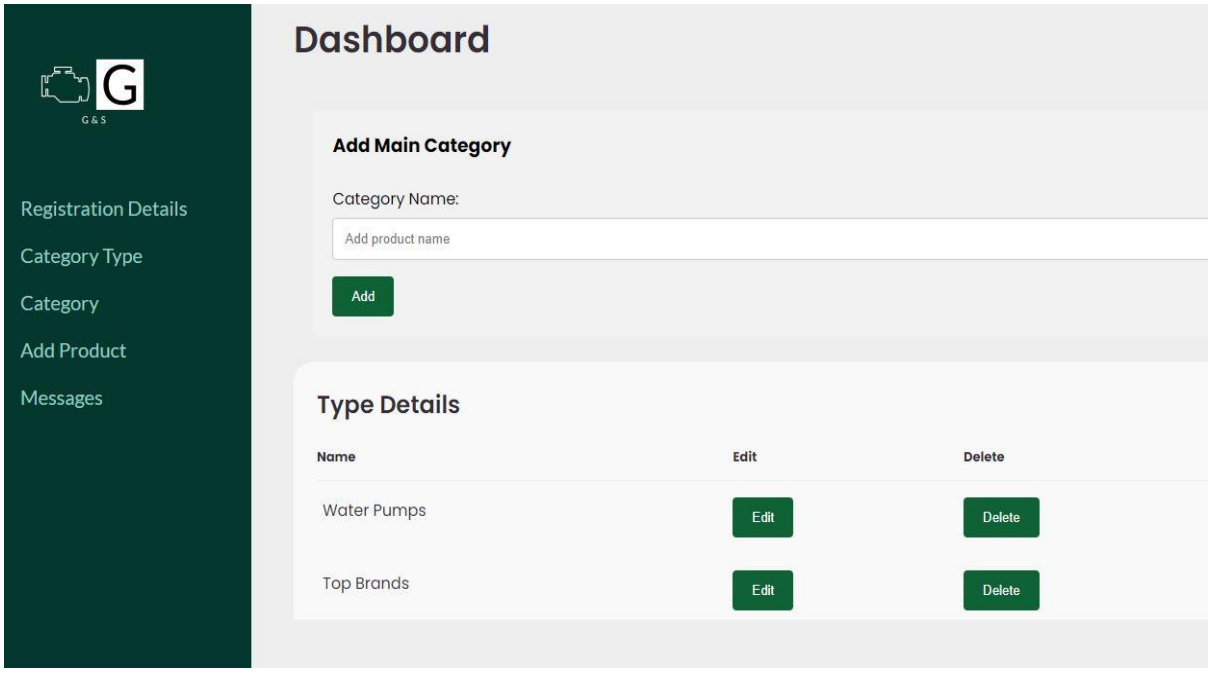


Product Home page

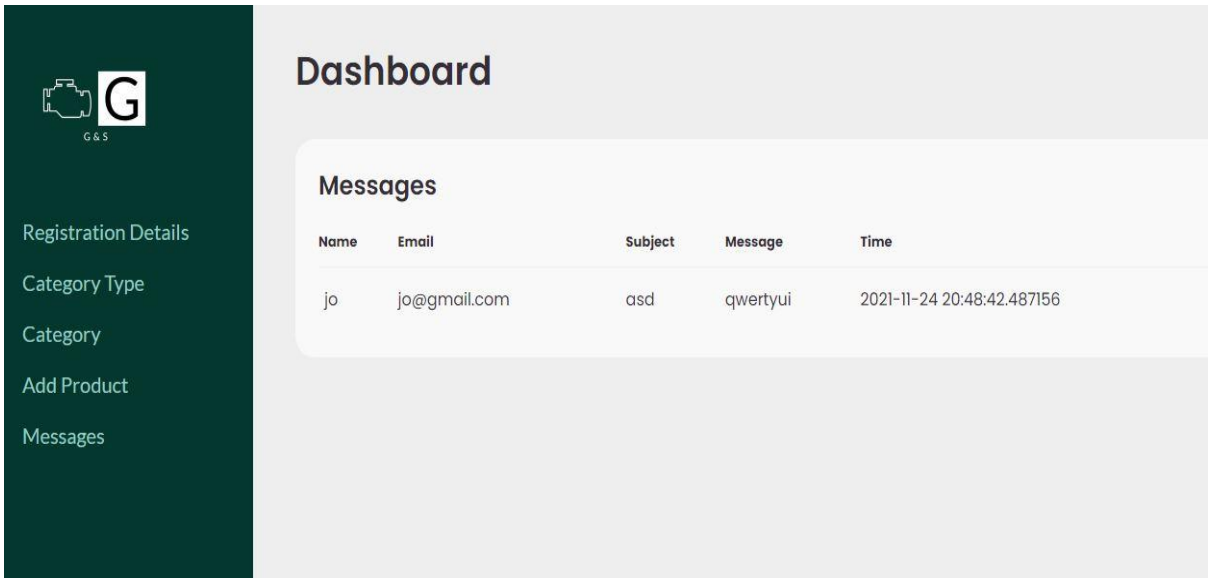


Admin Dashboard page

Add Category page



Feedback page



Registered users page