# E-MEDS ONLINE MEDICAL STORE

*Project Report Submitted By*

**ROSEMOL THOMAS**

**Reg.No: AJC17MCA-I049**

*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, **"E-MEDS ONLINE MEDICAL STORE"** is the bonafide work of **ROSEMOL THOMAS (Reg. No:AJC17MCA-I049)** 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. Rini Kurian**
**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 **"E-MEDS ONLINE MEDICAL STORE"** is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfillment 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:** **ROSEMOL THOMAS**

**KANJIRAPPALLY** **Reg. No: AJC17MCA-I049**

# ACKNOWLEDGEMENT

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

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr.Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project 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. Rini Kurian** for her inspiration and helping hand.

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

ROSEMOL THOMAS

# ABSTRACT

**E-Meds Online Medical Store** is an e-Medical initiative, which deals with the online transactions in making a medical store go online. It integrates the entire resources of a retail medical store into one integrated software application. This includes various facilities like viewing various products online and ordering them online. The main objective of this project is to provide a platform wherein one can access all the retail medical facilities online.

The primary point of building up this application is to supply the medications everywhere and to diminish the time utilization. Medication is given at your doorstep by the closest partner store. The remedy is required for requesting medication. According to the remedy, the client can look medication and valuable data. This application gives data to day by day utilization of drug. This application gives client login to the client. Also, administrator can get the every lapsed prescription data and he can ready to see all requests data of customers.

Customers can easily visit this site and register themself, by filling a registration form. Once a customer is registered he/she can login using their email and password. They can update their cart as per their need and make payment to the purchase. Provides searching facilities based on various factors. Admin provides all administrative support to ensure efficient operation of the software. This web application tracks all the information of users, medicinal products and payment. No formal knowledge is needed for the user to use this site. Thus by this all it proves it is user friendly.

# 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

**E-Meds Online Medical Store** is an e-Medical initiative, which deals with the online transactions in making a medical store go online. It integrates the entire resources of a retail medical store into one integrated software application. This includes various facilities like viewing various products online, comparing the prices of various commodities and ordering them online. The main objective of this project is to provide a platform wherein one can access all the retail medical facilities online.

Customers can easily visit this site and register themself, by filling a registration form. Once a customer is registered he/she can login using their email and password. They can update their cart as per their need and make payment to the purchase. Provides searching facilities based on various factors. Admin provides all administrative support to ensure efficient operation of the software. This web application tracks all the information of users, medicinal products and payment. No formal knowledge is needed for the user to use this site. Thus by this all it proves it is user friendly.

## 1.2 PROJECT SPECIFICATION

The proposed system is made to help the users for an easy and convenient way to buy medicines and medical equipments. We will also provide users to give feedbacks, about the interface and for improvements etc.
The system includes 2 modules. They are:

### 1. Admin Module

Admin has been granted full access with complete permissions towards the system. Maintain database is series of their duty for the functioning of the site. He can add and manage company, category and product to the database. He is essentially responsible for creating, deleting and modifying any product into the inventory. He can view the registered users and their details. This module gives all the administrative support to the web application.

### 2. End-Users Module

The user or further called as the end-users can be divided two categories:

**a. Not a registered user:**

They are only allowed to view the products and product details such company and price. They are limited to access for the site full potentials.

**b. Registered credentials:**

They are real users of the system. They can register their account and can use this registered credentials to login to the web application. They can view the products and product details such company and price. They are allowed to search the products and can add the product to their cart.

# 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

The existing Online Medical Store system is time consuming and requires more man power to function well. Secondly the scope of offline medical store is limited to local area and is available for fixed timing. All the data management involving product availability, searching, billing and

other report generation are done manually which indeed are very time consuming and also costs are high. It's not possible to get all the information or details as well as we can't satisfy the user through this process. 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

- Less convenient
- Human effort is needed.

## 2.4 PROPOSED SYSTEM

The proposed E-meds Online Medical Store system will completely revolutionize the industry. Searching of products, order placing, billing and product stock can be maintained by a single click. The order placed can be easily tracked at any time. The server side of the system has added the category, company, and product's details. The proposed system has the following requirements:

- System needs store information about new entry of customer.
- System need to maintain quality record.
- System need to keep the record of medicine.
- System need to update and delete the record.
- System also needs a search area.
- It also needs a security system.

## 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:

➢ **One for all:-**

Everything comes under single roof creates less vagueness about all and need not to have separate application for each services.

➢ **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. It will also provide data security as we are using the secured databases for maintaining the documents.

- **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 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 give an indication of the system is economically possible for development.

The cost of project, **E-meds-Online Medical 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. 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.

**E-meds**, GUI is simple so that users can easily use it. **E-meds** is simple enough so that no training is needed.

## 3.2 SYSTEM SPECIFICATION

### 3.2.2 Hardware Specification

Processor : Intel Core I3 and above

RAM : 512MB and Above

HDD : 40 GB and Above

### 3.2.3 Software Specification

Client on PC : Windows, Linux

Browser : Google Chrome, Microsoft Edge

Front End : HTML, CSS

Backend : MYSQL

Technologies used : JS, HTML5, AJAX, J Query, 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. PHP 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.

• **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. 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.

- **MySQL databases 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
- Statechart diagram
- Deployment diagram
- Component diagram
- Object 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 for the modeling of real-world objects and systems.

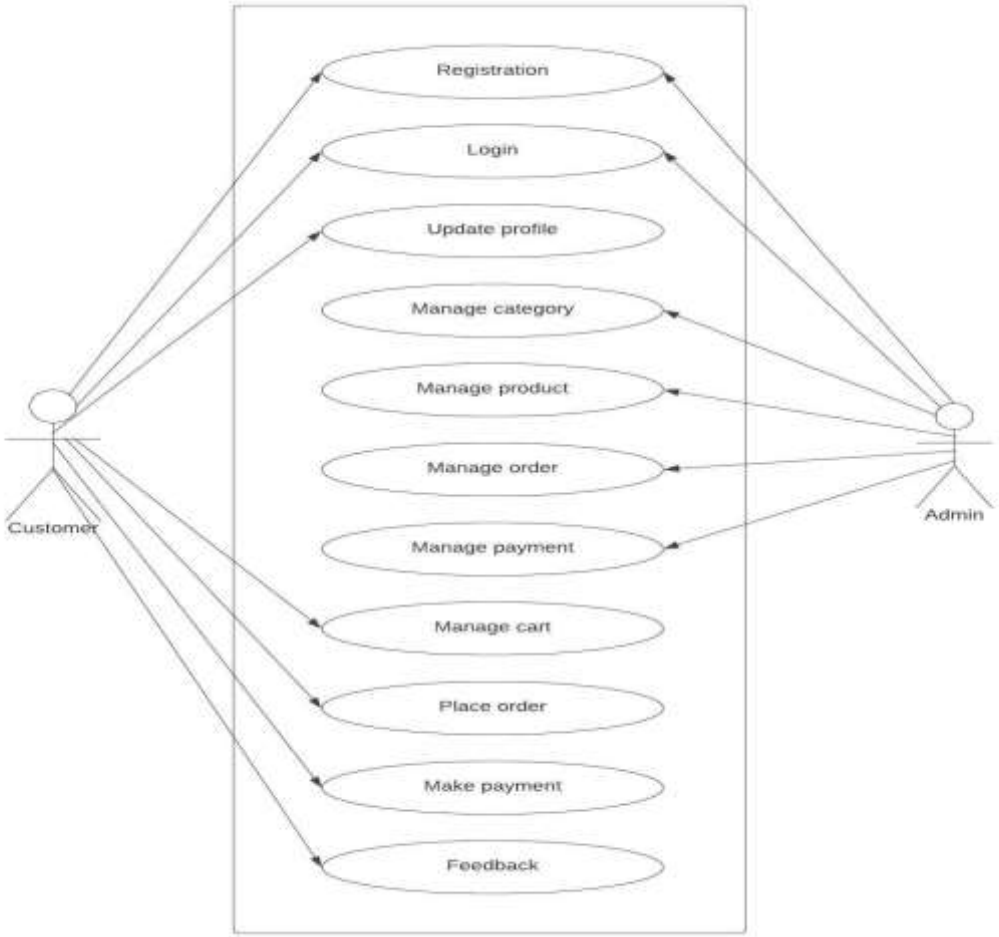
System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include 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.

Fig 1: Use case diagram for **E-meds Online Medical Store**




## 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. 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:

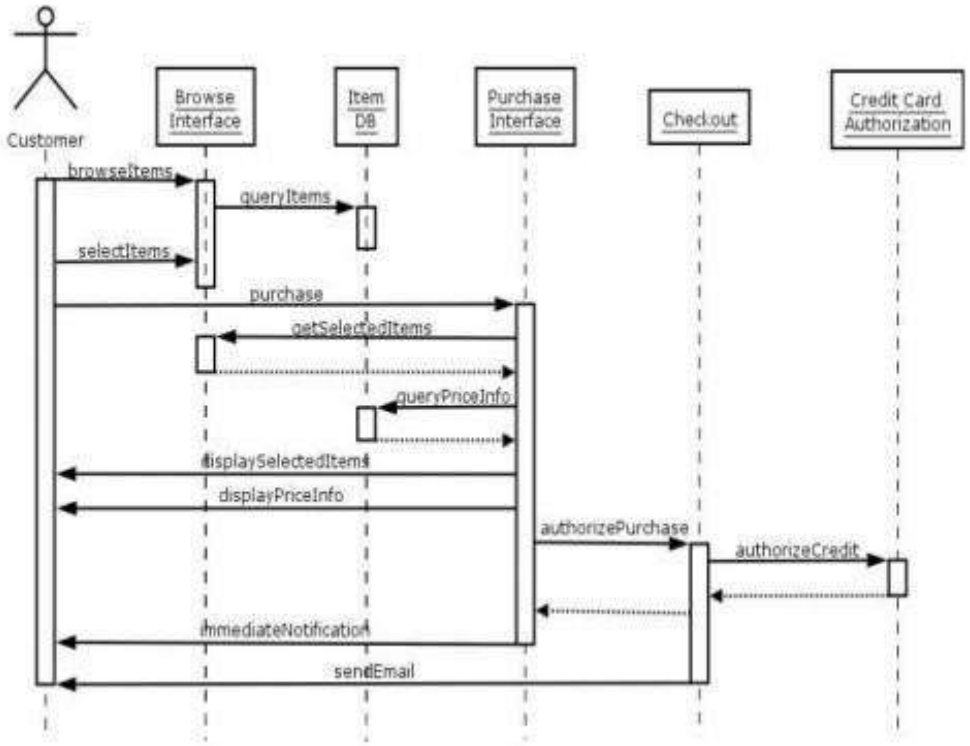
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

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.
- 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 for **E-meds Online Medical Store**




## 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 objectoriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

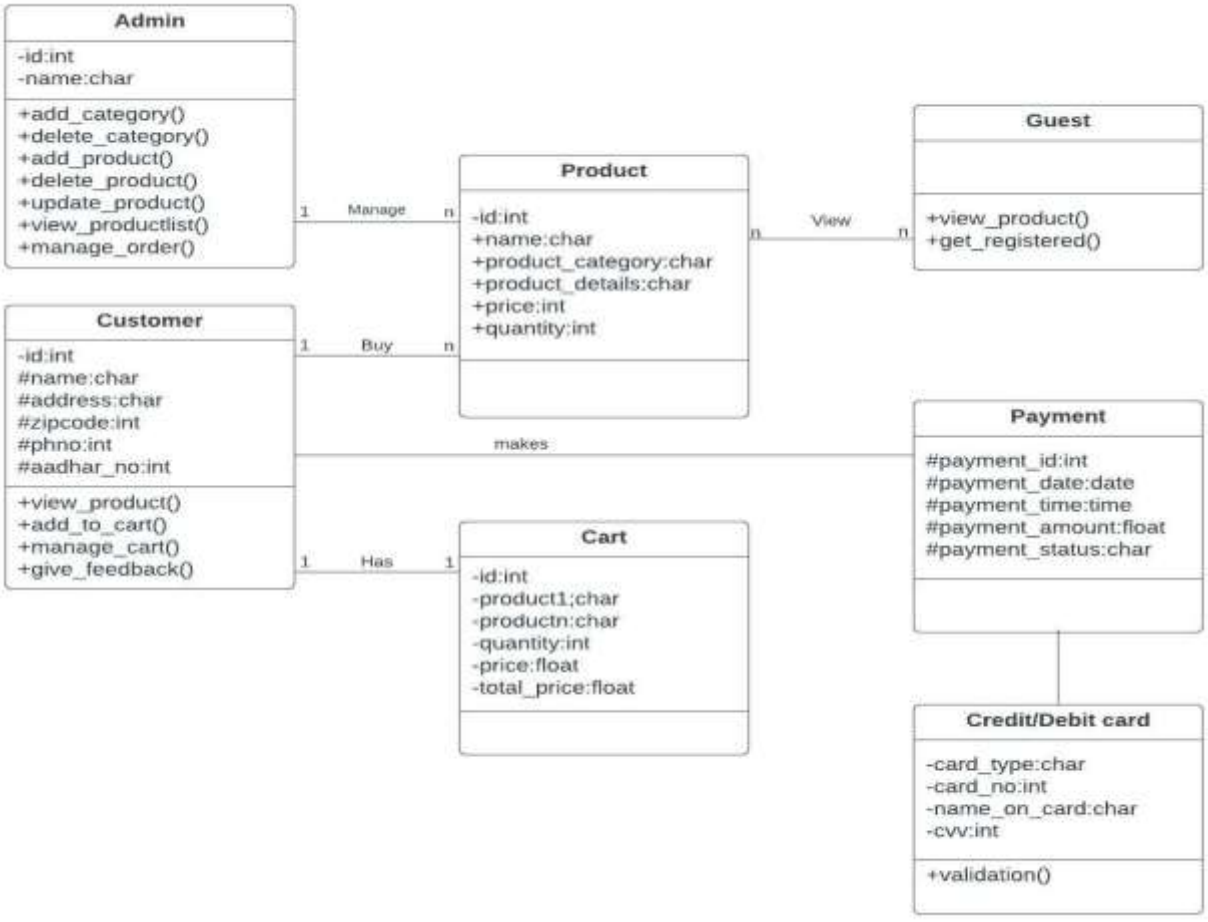
The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

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 **E-meds Online Medical Store**

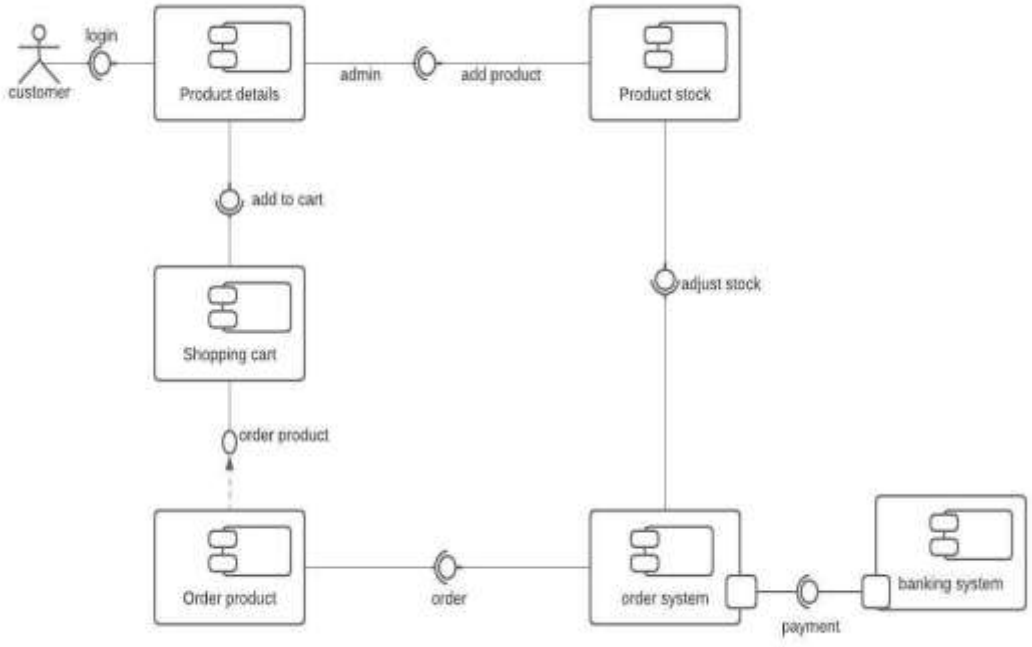


### 4.2.4 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? 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.

Fig 4: Component diagram for **E-meds Online Medical Store**



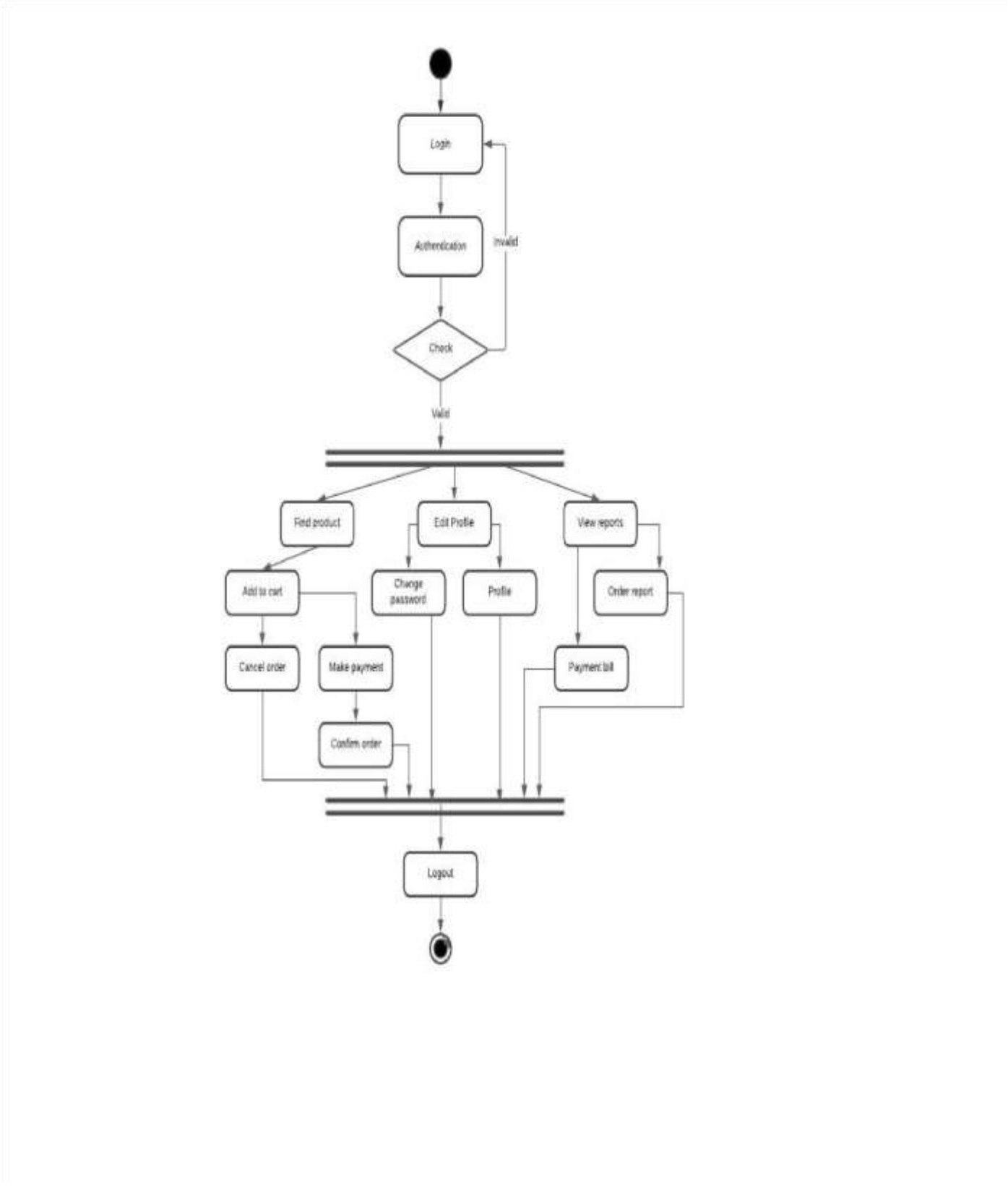
### 4.2.5 ACTIVITY DIAGRAM

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 is 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.

.

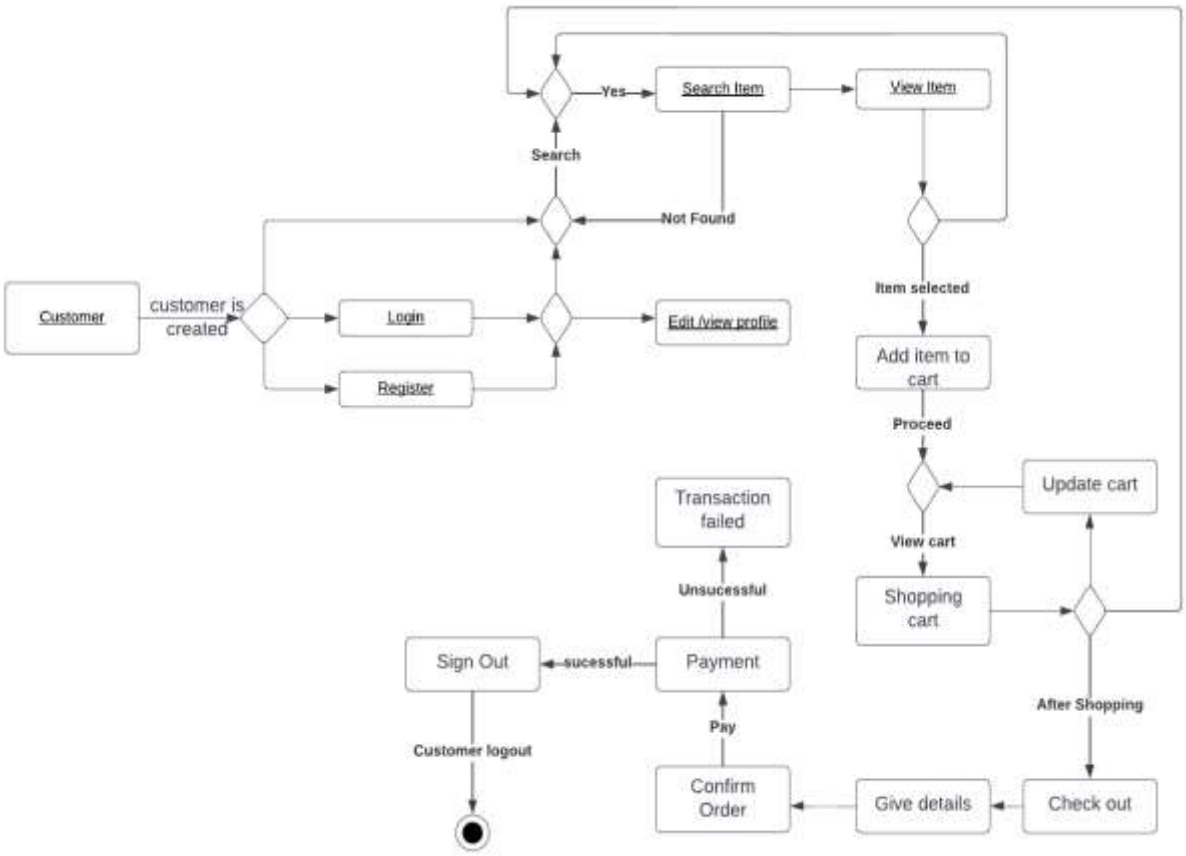
Fig 5: Activity diagram for **E-meds Online Medical Store**



### 4.2.6 STATE CHART DIAGRAM

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Activity diagram explained in the next chapter, is a special kind of a State chart diagram. State chart diagram defines the states, it is used to model the lifetime of an object. It is 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. Reactive systems can be defined as a system that responds to external or internal events.

Fig 6: State chart diagram for **E-meds Online Medical Store**



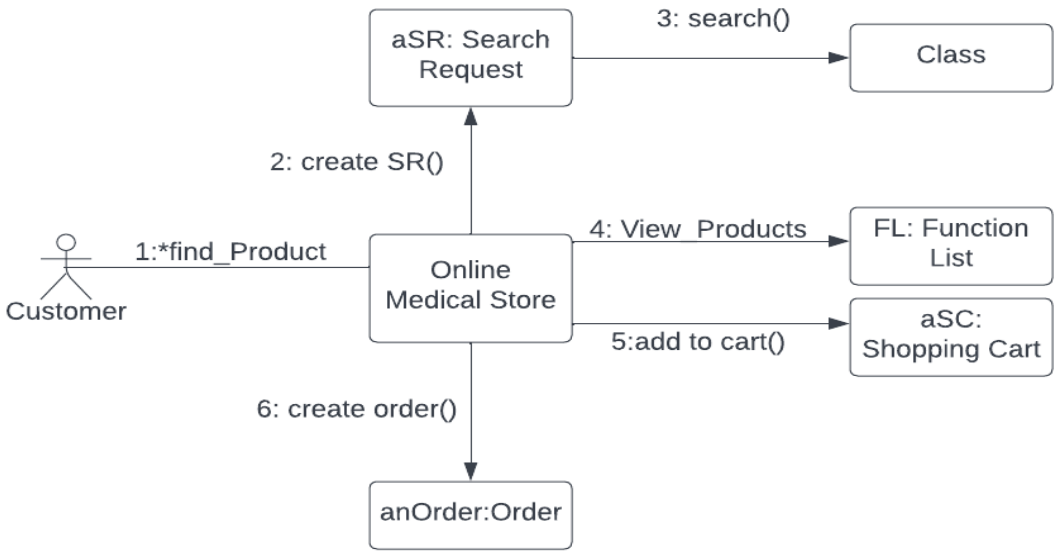

### 4.2.7 COLLABORATION DIAGRAM

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. These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams. Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read and use efficiently. Additionally, collaboration diagrams typically exclude descriptive information, such as timing.

Fig 7: Collaboration diagram for **E-meds Online Medical Store**




### 4.2.8 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. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

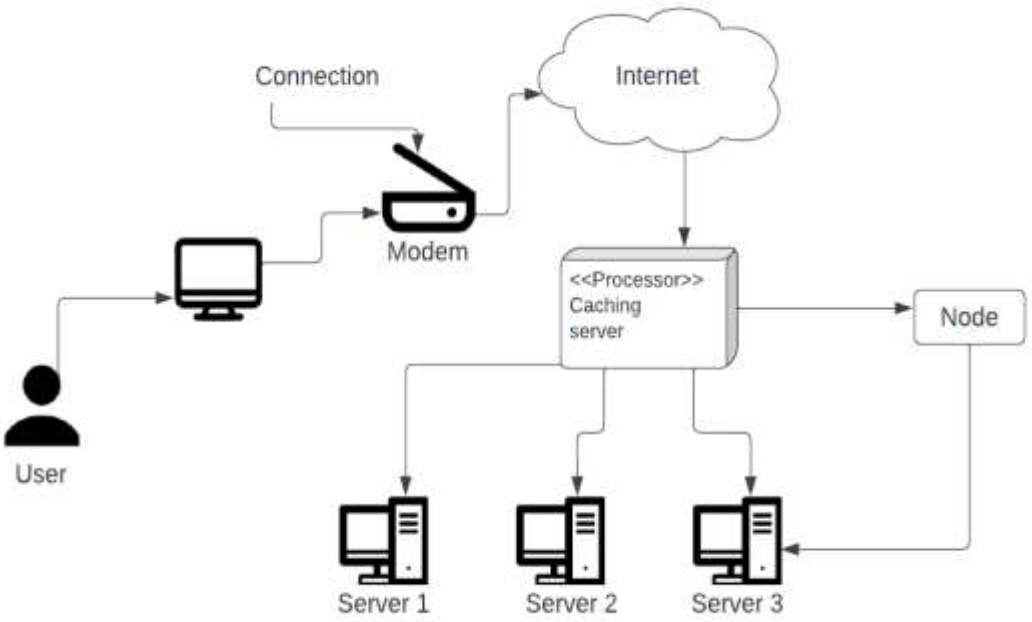
The purpose of deployment diagrams can be described as −

- 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 −

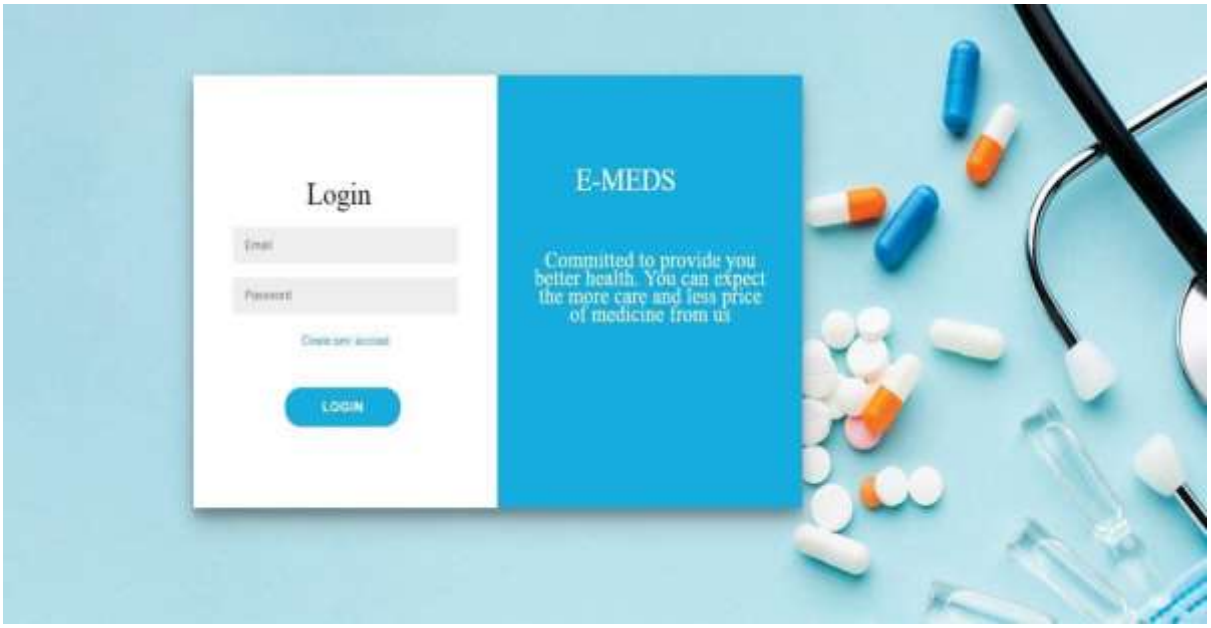
- Performance
- Scalability
- Maintainability
- Portability

Fig 8: Deployment diagram for **E-meds Online Medical Store**

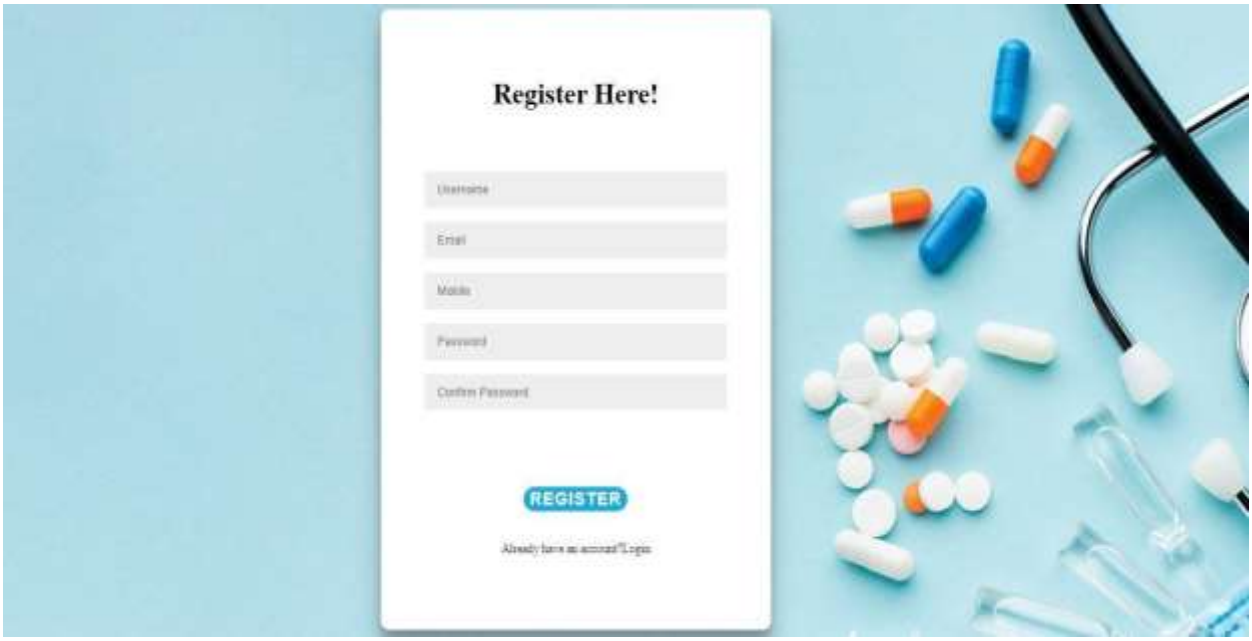


## 4.3 USER INTERFACE DESIGN USING FIGMA

### User Login



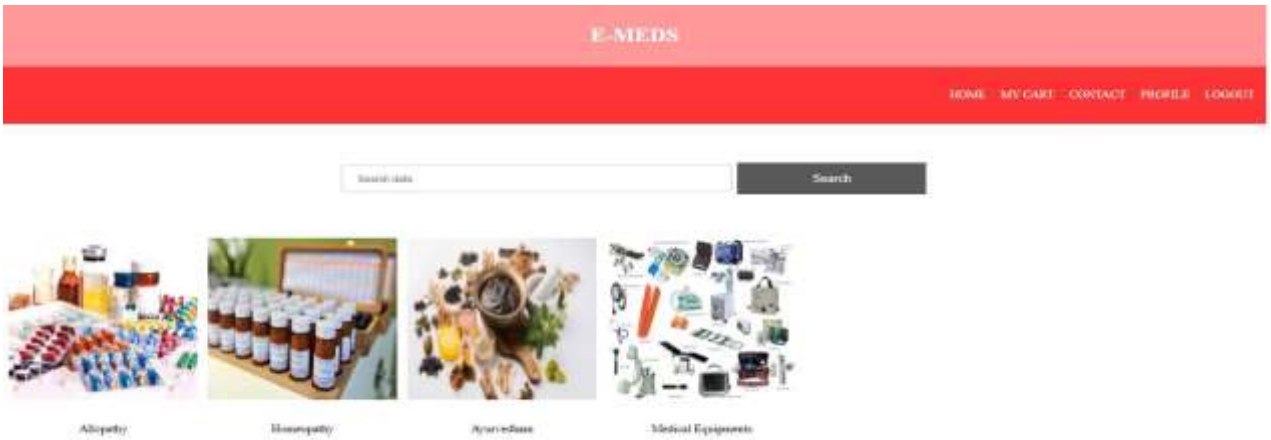

### User Registration



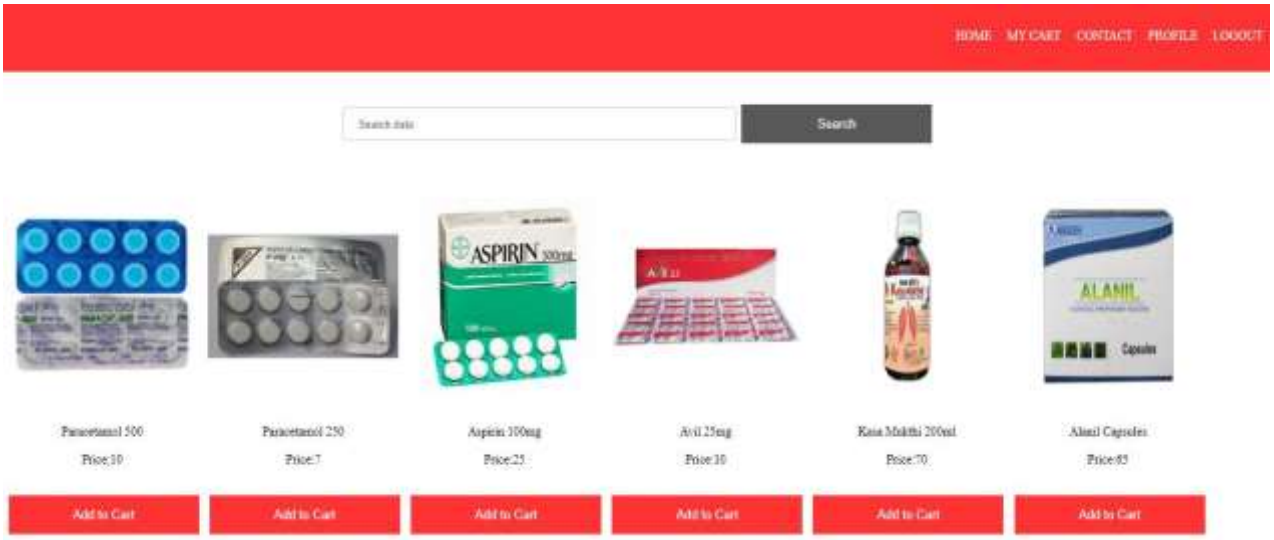
## Add Category


## Add Product


## User Home



## Products




## Single product

## 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 key 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 attributes of the relation is fully dependent on its primary key alone.

**Third Normal Form**

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

## TABLE DESIGN

### Table No 01: tbl_login

Primary Key: login_id

| Field Name | Type | Size | Description |
|---|---|---|---|
| LOGIN_ID | Int | 10 | Primary key of registration table |
| uname | Varchar | 30 | Username |
| email | Varchar | 30 | Email-id of the user |
| ph_no | Int | 30 | Mobile no. of the user |
| password | Varchar | 30 | Password |
| status | Varchar | 10 | To check whether the user is available or not |

### Table No 02: tbl_category

Primary Key: cat_id

| Field Name | Type | Size | Description |
|---|---|---|---|
| cat_id | Int | 10 | Primary key of Category table |
| cat_name | Varchar | 50 | Category name |
| img | Varchar | 255 | Category image |

**Table No 03 : tbl_product**

Primary Key : prod_id

Foreign Key : cat_id

| Field Name | Type | Size | Description |
|---|---|---|---|
| prod_id | Int | 10 | Primary key of product table |
| cat_id | Int | 10 | Category Id Foreign key of product table |
| prod_name | Varchar | 50 | Product name |
| details | Varchar | 100 | Details of the product |
| price | Int | 50 | Price of product |
| img | Varchar | 10 | Product image |

**Table No 04 : tbl_cart**

Primary Key : cart_id

Foreign Key : LOGIN_ID

| Field Name | Type | Size | Description |
|---|---|---|---|
| id | Int | 10 | Primary key of cart table |
| LOGIN_ID | Int | 10 | User login id Foreign key of cart table |
| pname | Varchar | 50 | Product name |
| price | Varchar | 100 | Product price |

| | | | |
|---|---|---|---|
| image | Varchar | 255 | Product image |
| quantity | Int | 11 | Product quantity |

**Table No 05 : tbl_feedback**

Primary Key : fb_id

Foreign Key : LOGIN_ID

| Field Name | Type | Size | Description |
|---|---|---|---|
| fb_id | Int | 11 | Primary key of feedback table |
| LOGIN_ID | Int | 11 | User login Id<br>Foreign key of feedback table |
| message | Varchar | 200 | User feedback |

**Table No 06 : tbl_payment**

Primary Key : id

| Field Name | Type | Size | Description |
|---|---|---|---|
| id | Int | 11 | Primary key of payment table |
| name | Varchar | 50 | User name |
| amount | Varchar | 11 | Total amount |
| payment_status | Varchar | 50 | Transaction status |
| payment_id | Varchar | 100 | Transaction id |

| | | | |
|---|---|---|---|
| added_on | Datetime | | Transaction date |

**Table No 07 : tbl_profile**

Primary Key : pro_id

Foreign Key : LOGIN_ID

| Field Name | Type | Size | Description |
|---|---|---|---|
| pro_id | Int | 11 | Primary key of profile table |
| LOGIN_ID | Int | 11 | User login id<br>Foreign key of profile table |
| gender | Varchar | 15 | User's gender |
| adrs | Varchar | 500 | Address of user |
| city | Varchar | 100 | User's city |
| state | Varchar | 50 | State of user |
| pin | Varchar | 10 | Postal code of user |
| aadhar | Varchar | 30 | Aadhar no. of user |

# 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 terms 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 appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are 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

**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.

**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.

**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.

### 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.

### Selenium Testing

Selenium is one of the most widely used open-source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages. Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven**,** Jenkins**,** & Docker to achieve continuous testing. It can also be integrated with tools such as TestNG**,** & JUnit for managing test cases and generating reports.

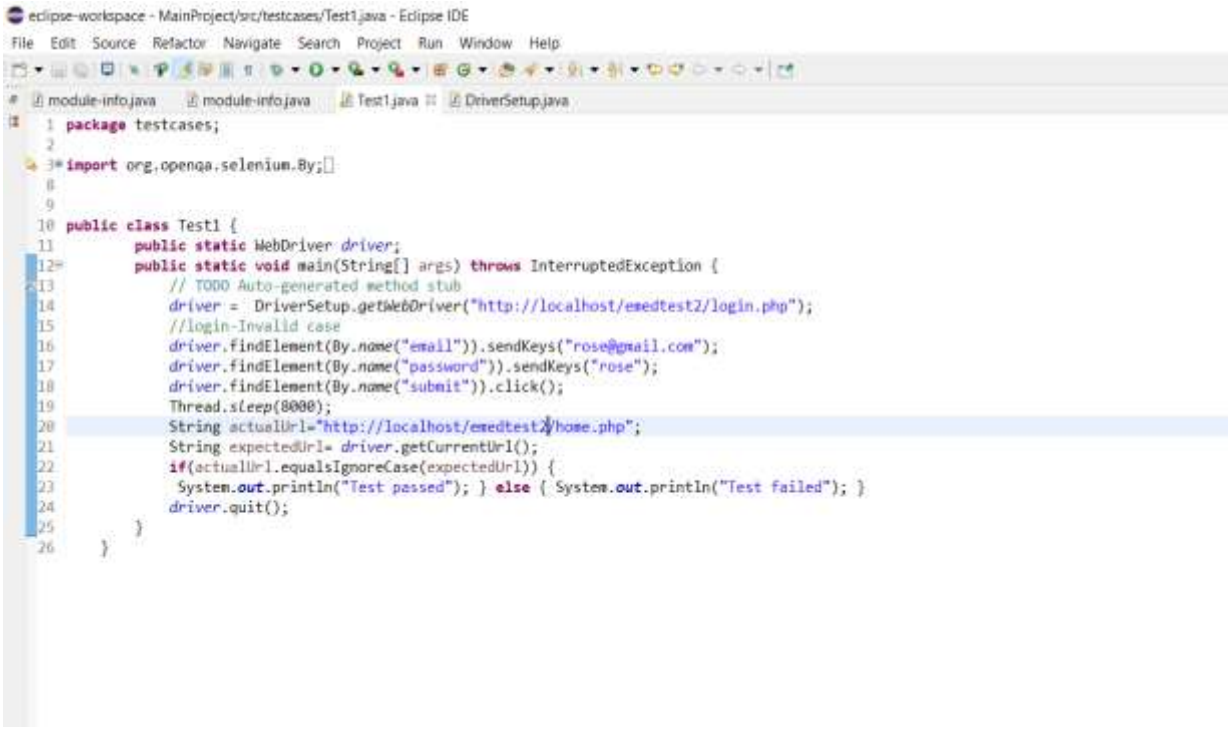
**Test case 1**

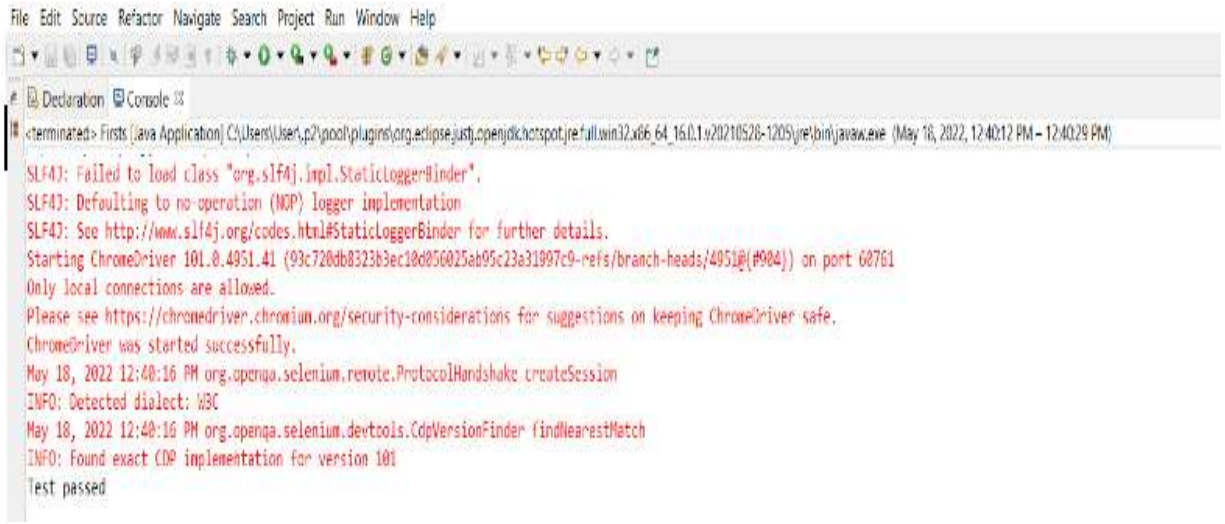
<table>
<tr><td colspan="6">Project Name: E-MEDS Online Medical Store</td></tr>
<tr><td colspan="6">Login Test Case</td></tr>
<tr><td colspan="3">Test Case ID: Fun_1</td><td colspan="3">Test Designed By: Rosemol Thomas</td></tr>
<tr><td colspan="3">Test Priority (Low/Medium/High): High</td><td colspan="3">Test Designed Date: 22-05-2022</td></tr>
<tr><td colspan="3">Module Name: Login Screen</td><td colspan="3">Test Executed By: Ms. Rini Kurian</td></tr>
<tr><td colspan="3">Test Title: Verify login with valid email and password</td><td colspan="3">Test Execution Date: 23-05-2022</td></tr>
<tr><td colspan="3">Description: Test the Login Page</td><td colspan="3"></td></tr>
<tr><td colspan="6">Pre-Condition: User has valid email id and password</td></tr>
<tr><td>Step</td><td>Test Step</td><td>Test Data</td><td>Expected Result</td><td>Actual Result</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1</td><td>Navigation to Login Page</td><td></td><td>Login Page should be displayed</td><td>Login page displayed</td><td>Pass</td></tr>
<tr><td>2</td><td>Provide Valid Email Id</td><td>User Name: rose@gmail .com</td><td rowspan="3">User should be able to Login</td><td rowspan="3">User Logged in and navigated to Subadmin Dashboard with records</td><td rowspan="3">Pass</td></tr>
<tr><td>3</td><td>Provide Valid Password</td><td>Password: rose</td></tr>
<tr><td>4</td><td>Click on Sign In button</td><td></td></tr>
<tr><td>5</td><td>Provide Invalid Email Id or password</td><td>Email Id: user@gmail. Com<br>Password: User12345</td><td rowspan="3">User should not be able to Login</td><td rowspan="3">Message for enter valid email id or password displayed</td><td rowspan="3">Pass</td></tr>
<tr><td>6</td><td>Provide Null Email Id or Password</td><td>Email Id: null<br>Password: null</td></tr>
<tr><td>7</td><td>Click on Sign In button</td><td></td></tr>
<tr><td colspan="6">Post-Condition: User is validated with database and successfully login into account.</td></tr>
</table>

## 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 {
//TODOAuto-generatedmethodstubdriver=
DriverSetup.getWebDriver("http://localhost/emedtest2/Login/Login.php");
//login-Invalid case
driver.findElement(By.name("email")).sendKeys("rose@gmail.com");
driver.findElement(By.name("psw")).sendKeys("rose");
driver.findElement(By.name("submit")).click();
Thread.sleep(8000);
String actualUrl="http://localhost/emedtest2/home.php"; String
expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed"); } else { System.out.println("Test nfailed"); }
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
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/emedtest2/login.php");
//login-Invalid case
driver.findElement(By.name("email")).sendKeys("rose@gmail.com");
driver.findElement(By.name("password")).sendKeys("rose");
driver.findElement(By.name("submit")).click();
Thread.sleep(8000);
String actualUrl="http://localhost/emedtest2/home.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> First1 [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 (93c720db8323b3ec10d850025ab95c23a31997c9-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 case 2**

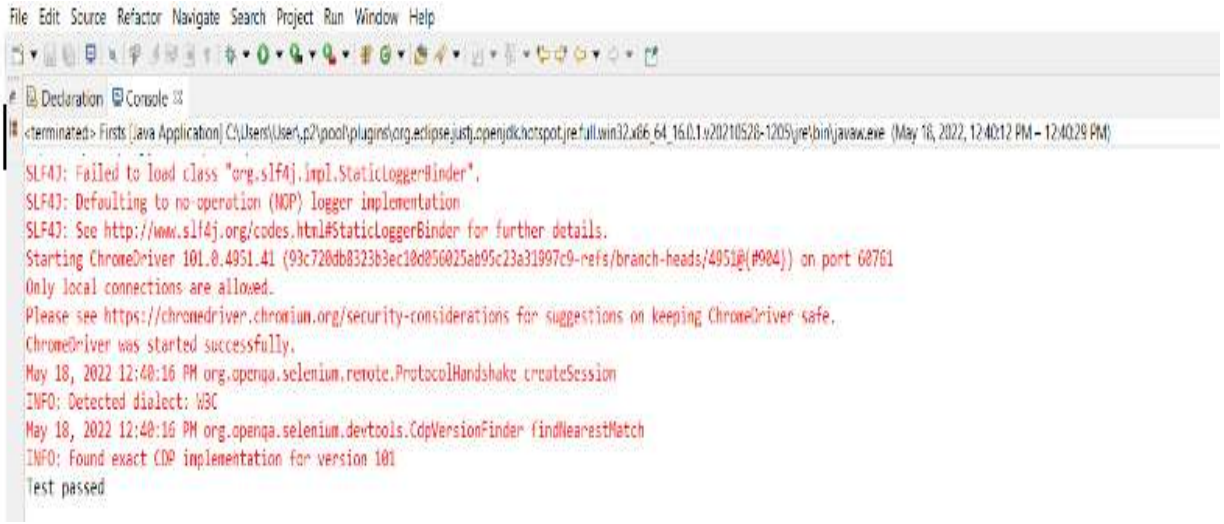
| **Project Name: Online Medical store** | | | | | |
|---|---|---|---|---|---|
| **Update Test Case** | | | | | |
| **Test Case ID:**updateProfile | | | **Test Designed By: Rosemol Thomas** | | |
| **Test Priority(Low/Medium/High):**High | | | **Test Designed Date:** 22-05-2022 | | |
| **Module Name**:Login Screen | | | | | |
| **Test Title :** Update house details | | | **Test Execution Date:** 23-05-2022 | | |
| **Description:** Login to system updateprofile, 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 LoginPage | | Login Page shouldbe. | Login page displayed | Pass |
| 2 | ProvideValid Username | User Name: rose | User should be able to Login | User Logged in and navigated to user home page | Pass |
| 3 | Provide Valid Password | Password: rose | | | |
| 4 | Click on Login button | | | | |
| 5 | Provide user informations | Input profile details | User will be redirected to home page | User will be redirected to home page | Pass |
| 7 | Click on Update button | | | | |
| 8 | Provide invalid informations | Input invalid house details | User will be redirected to home page | User will be stay on that page showing the error messsage | Pass |
| 9 | Click on Add 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/emedtest2/login.php");
driver.findElement(By.name("login")).click();
driver.findElement(By.name("email")).sendKeys("rose@gmail.com");
driver.findElement(By.name("password")).sendKeys("rose");
driver.findElement(By.name("submitButton")).click();
driver.get("http://localhost/emedtest2/profile.php");
driver.findElement(By.name("name")).sendKeys("Rose");
driver.findElement(By.name("phno")).sendKeys("7558943452");
driver.findElement(By.name("adrs")).sendKeys("padickal");
driver.findElement(By.name("city")).sendKeys("kumily");
driver.findElement(By.name("state")).sendKeys("kerala");
driver.findElement(By.name("pin")).sendKeys("685535");
driver.findElement(By.name("aadhar")).sendKeys("4382412048832593");
driver.findElement(By.name("add")).click();
String  actualUrl="http://localhost/emedtest2/login.php";
expectedUrl= driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
System.out.println("Test passed");
} else {
System.out.println("Test failed");
}
}
}
```

## Output


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 (93c720db8323b3ec10d050025ab95c23a31997c9-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.

**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. 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

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge about the web development. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

The existing system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility to buy medicines and medical equipments online. We provide the users to view product details and make payment for the purchase. Admin can add the category, company and product to the website, also he can view the customer information. Searching of products, order placing, billing can be maintained by a single click. The order placed can be easily tracked at any time. The server side of the system has added the category, company, and product's details.

## 7.2 FUTURE SCOPE

- Communication through chats
- Transferring data with high security
- Managing stock
- Create monthly/ yearly information

# 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.
- Pankaj Jalote, “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 8

# APPENDIX

## 8.1 SAMPLE CODE

### cart.php

```
<?php
  include('dbconn.php');
  session_start();
  $cname=$_SESSION['name'];
  $q="SELECT * FROM `tbl_login` WHERE email='$cname'";
  $data=mysqli_query($con,$q);
  if($res=mysqli_fetch_assoc($data))
  {
  $ker=$res['LOGIN_ID'];
  }
  if(isset($_POST['update_cart'])){
  $update_quantity = $_POST['cart_quantity'];
  $update_id = $_POST['cart_id'];
  mysqli_query($con, "UPDATE `tbl_cart` SET quantity = '$update_quantity' WHERE id =
  '$update_id'") or die('query failed');
  $message[] = 'cart quantity updated successfully!';
  }
  if(isset($_GET['remove'])){
  $remove_id = $_GET['remove'];
  mysqli_query($con, "DELETE FROM `tbl_cart` WHERE `id` = '$remove_id'") or die('query
  failed');
  header('location:cart.php');
  }
  if(isset($_GET['delete_all'])){
  mysqli_query($con, "DELETE FROM `tbl_cart` WHERE LOGIN_ID = '$ker'") or die('query
  failed');
  header('location:cart.php');
  }
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"content="width=device-width,initial-scale=1.0">
  <title>Cart</title>
```

```
</head>
<body>
    <div class="main">
    <div class="header">
    <h1>E-MEDS</h1>
    </div>
    <div class="navbar">
    <ul>
    <li><a href="login.php">LOGOUT</a></li>
    <li><a href="profile.php">PROFILE</a></li>
    <li><a href="contact.php">CONTACT</a></li>
    <li><a href="mycart.php">MY CART</a></li>
    <li><a href="home.php">HOME</a></li>
    </ul>
    </div>
    <div class="shopping-cart">
    <h1 class="heading">CART</h1>
    <table>
    <thead>
    <th>IMAGE</th>
    <th>NAME</th>
    <th>PRICE</th>
    <th>QUANTITY</th>
    <th>TOTAL PRICE</th>
    <th>ACTION</th>
    </thead>
    <tbody>
    <?php
    $cart_query = mysqli_query($con, "SELECT * FROM `tbl_cart` WHERE `LOGIN_ID` =
    '$ker'") or die('query failed');
    $grand_total = 0;
    if(mysqli_num_rows($cart_query) > 0){
    while($fetch_cart = mysqli_fetch_assoc($cart_query)){
    ?>
    <tr>
    <td><img src="image/<?php echo $fetch_cart['image']; ?>" width="50px"
    height="50px"></td>
    <td><?php echo $fetch_cart['pname']; ?></td>
    <td>₹<?php echo $fetch_cart['price']; ?>/-</td>
```

```
<td>
<form action="" method="post">
<input type="hidden" name="cart_id" value="<?php echo $fetch_cart['id']; ?>">
<input type="number" min="1" name="cart_quantity" value="<?php echo
 $fetch_cart['quantity']; ?>">
<input type="submit" name="update_cart" value="update" class="option-btn">
</form>
</td>
<td>₹<?php echo $sub_total = ($fetch_cart['price'] * $fetch_cart['quantity']); ?>/-</td>
<td><a href="cart.php?remove=<?php echo $fetch_cart['id']; ?>" class="delete-btn"
 onclick="return confirm('remove item from cart?');">remove</a></td>
</tr>
<?php
$grand_total += $sub_total;
}
}else{
echo '<tr><td style="padding:20px; text-transform:capitalize;" colspan="6">no item
 added</td></tr>';
}
?>
<tr class="table-bottom">
<td colspan="4" >Grand Total :</td>
<td>₹<?php echo $grand_total; ?>/-</td>

<td><a href="cart.php?delete_all" onclick="return confirm('delete all from cart?');"
 class="delete-btn <?php echo ($grand_total > 1)?":'disabled'; ?>">delete all</a></td>
</tr>
</tbody>
</table>
<br><br>
<div class="container">
<form>
<div class="row">
<div class="col-50">
<h3>Billing Address</h3>
<?php
$q="SELECT * FROM `tbl_login` WHERE email='$cname'";
$data=mysqli_query($con,$q);
if($res=mysqli_fetch_assoc($data))
{
```

```
$ker=$res['LOGIN_ID'];
$sql="SELECT * FROM `tbl_profile` WHERE LOGIN_ID='$ker'";
$data3=mysqli_query($con,$sql);
if($res1=mysqli_fetch_assoc($data3))
{
?>
<label for="fname" ><i class="fa fa-user"></i> Full Name</label>
<input type="text" id="name" name="name" placeholder="" value="<?php echo
$res['uname'];?>">
<label for="adr"><i class="fa fa-address-card-o"></i> Address</label>
<input type="text" id="adr" name="address" placeholder="" value="<?php echo
$res1['adrs'];?>"required>
<label for="city"><i class="fa fa-institution"></i> City</label>
<input type="text" id="city" name="city" placeholder="" value="<?php echo $res1['city'];?>"
required>
<div class="row">
<div class="col-25">
<label for="state">State</label>
<input type="text" id="state" name="state" placeholder="" value="<?php echo
$res1['state'];?>" required>
</div>
<div class="col-25">
<label for="zip">Pincode</label>
<input type="text" id="zip" name="zip" placeholder="" value="<?php echo $res1['pin'];?>"
required>
</div>
</div>
<?php }
}
?>
</div>
</div>
</div>
<center><label for="zip">Total Amount</label>
<input type="text" name="amt" id="amt" value="<?php echo $grand_total; ?>" disabled
style="width:25%;" >
</center>
<div class="cart-btn">
<input type="button" name="btn" class="btn" id="btn" value="Pay Now"
onclick="pay_now()"/>
```

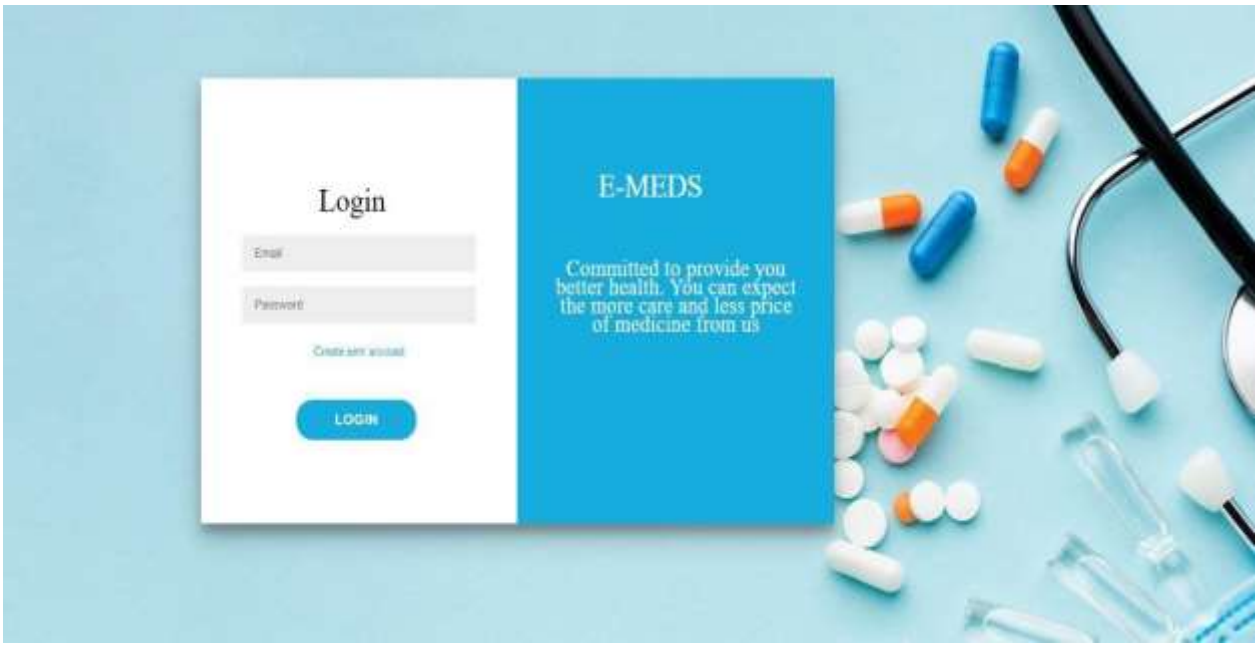
```
</div>
</form>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script src="https://checkout.razorpay.com/v1/checkout.js"></script>
<script>
function pay_now(){
var firstName=document.getElementById("name");
if (firstName.value===""){
alert("Please enter your name");
firstName.focus();
return false;
}
var postcode=document.getElementById("zip");
if (postcode.value.length!=6  || isNaN(postcode.value)){
alert("Please enter 6 digit pincode");
postcode.focus();
return false;
}
var name=jQuery('#name').val();
var amt=jQuery('#amt').val();
jQuery.ajax({
type:'post',
url:'payment_process.php',
data:"amt="+amt+"&name="+name,
success:function(result){
var options = {
"key": "rzp_test_sNi1Kb8hhLsH46",
"amount": <?php echo $grand_total; ?> *100,
"currency": "INR",
"name": "Acme Corp",
"description": "Test Transaction",
"image": "https://image.freepik.com/free-vector/logo-sample-text_355-558.jpg",
"handler": function (response){
jQuery.ajax({
type:'post',
url:'payment_process.php',
```

```
    data:"payment_id="+response.razorpay_payment_id,
    success:function(result){
    window.location.href="thank_you.php";
    }
    });
    }
    };
    var rzp1 = new Razorpay(options);
    rzp1.open();
    }
    });
    }
    </script>
</body>
</html>
<script type = "text/javascript" >
function preventBack() { window.history.forward(); }
setTimeout("preventBack()", 0);
window.onunload = function () { null };
</script>
```

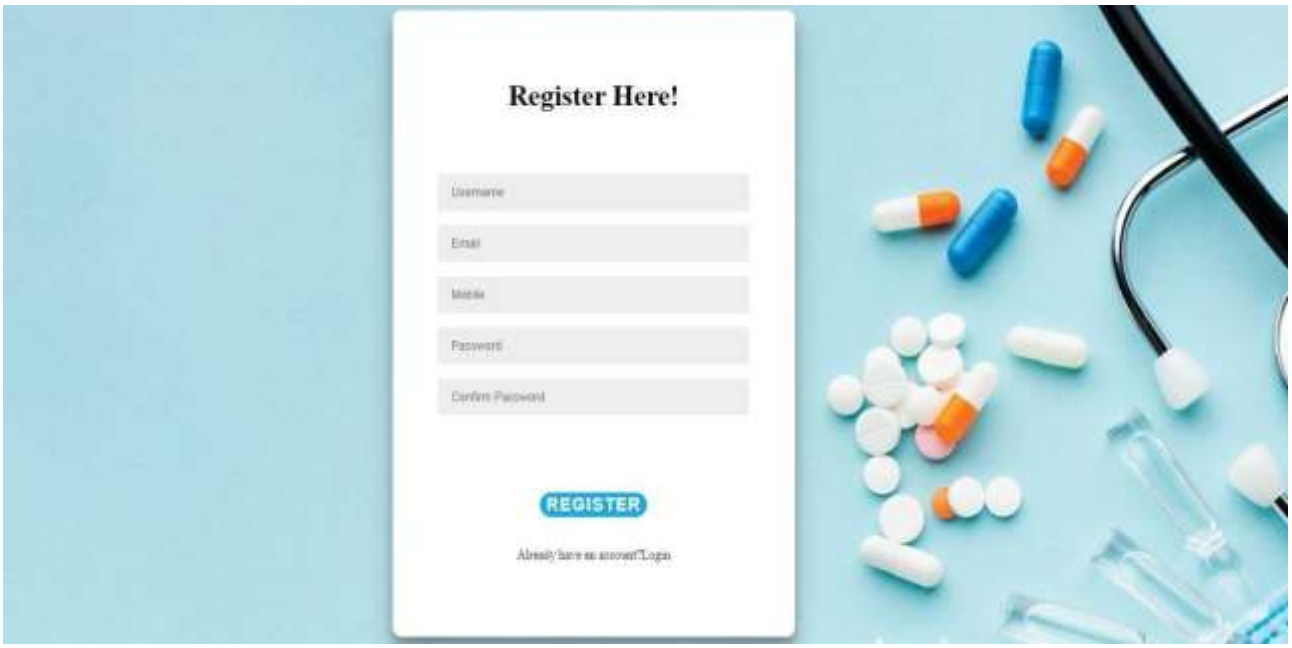
## 8.2 SCREEN SHOTS

### CUSTOMER PAGES

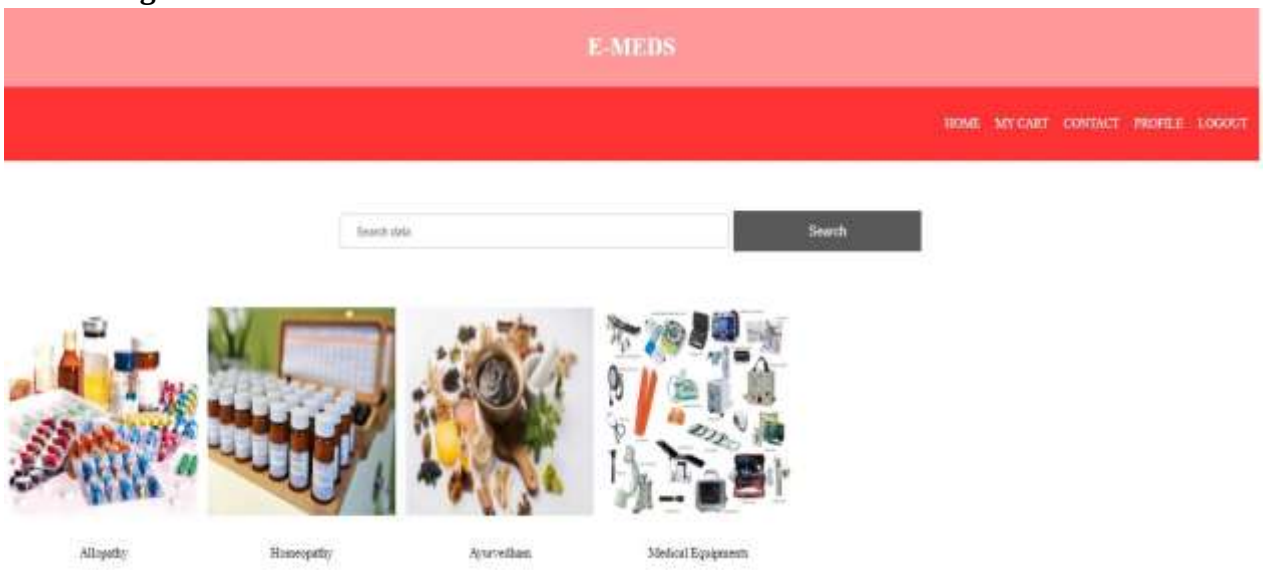
#### User Login



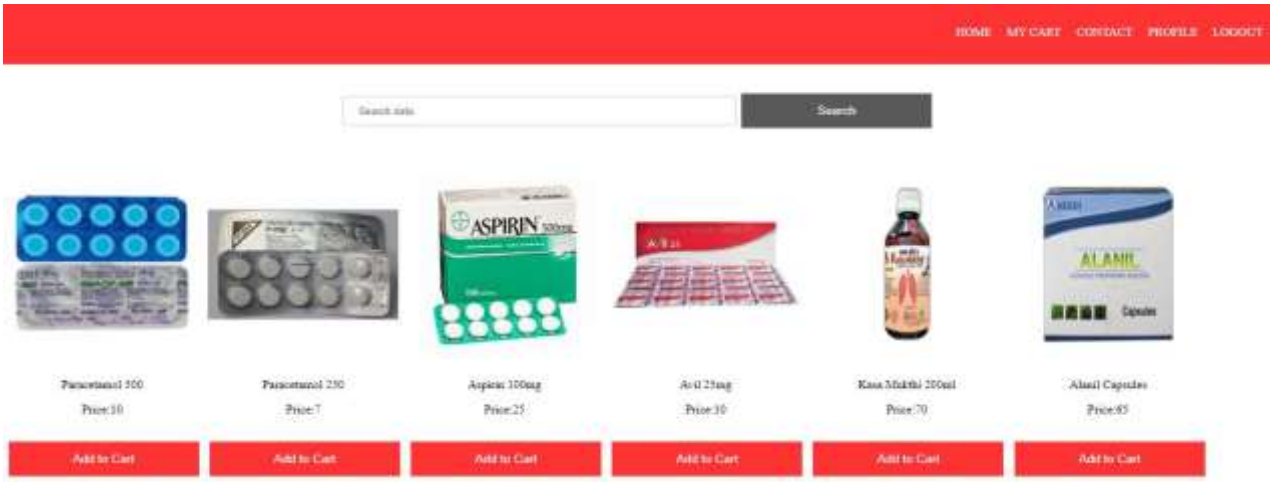

#### User Register



## Home Page




## Products page

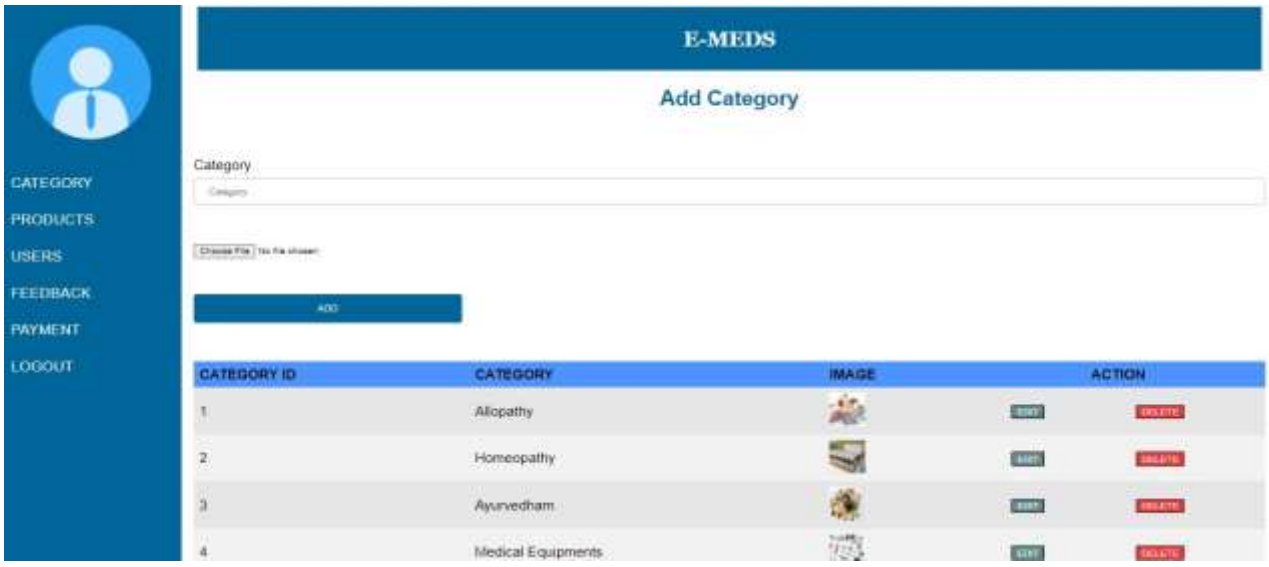



## Cart

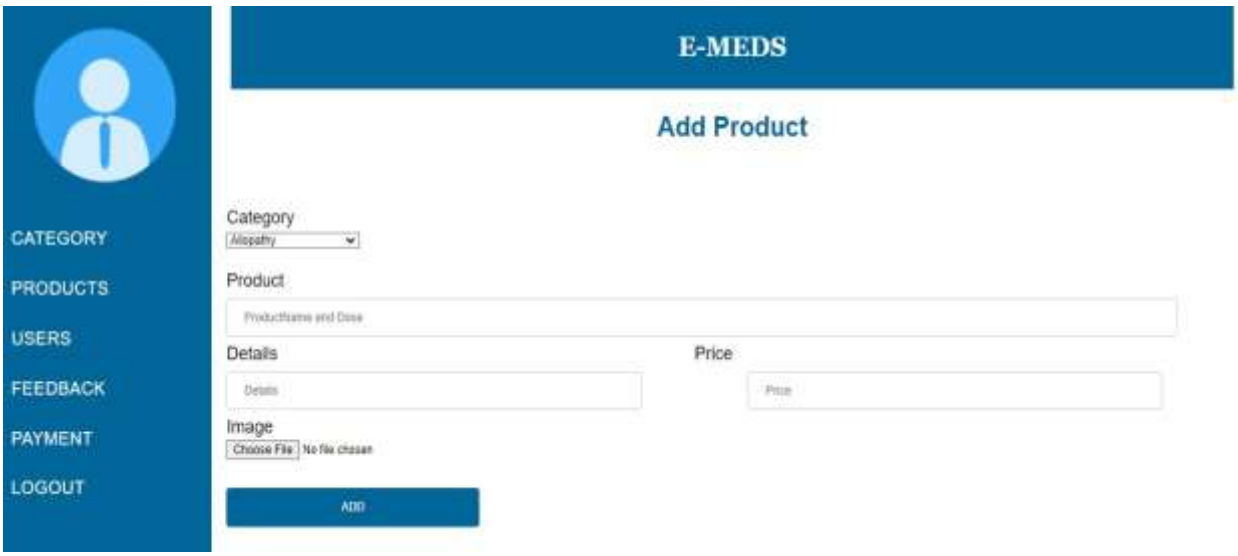
### Billing

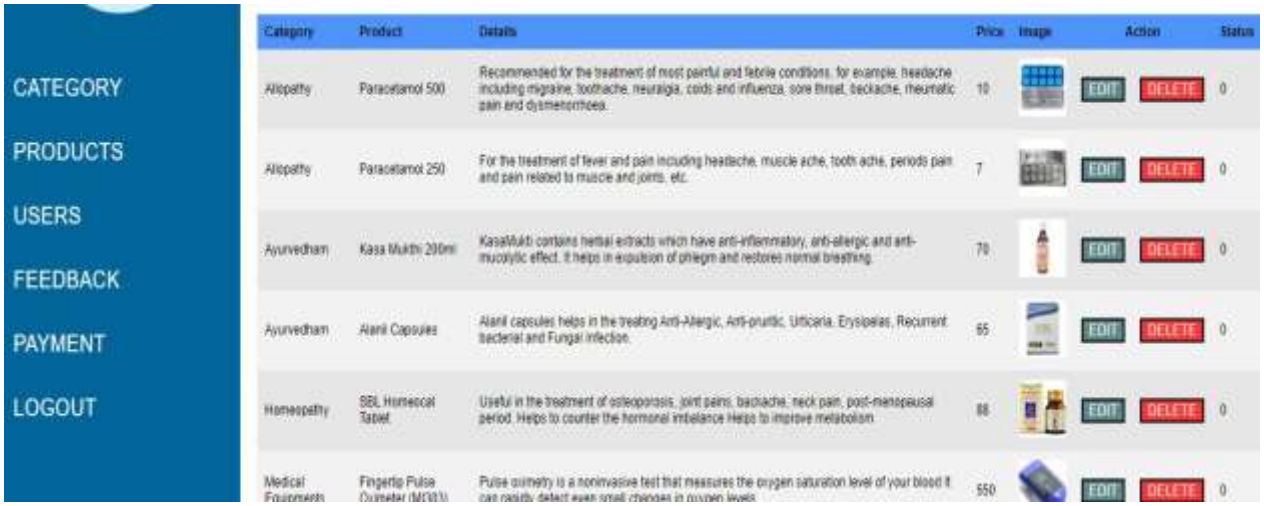

## ADMIN PAGES

### Add and View Category



## Add and View Product



## Update Product