E-Medicare

Group-1: Team Members

S.No	Employee Name	Employee ID
1.	Abhijit Bandu Jawale	2489569
2.	Abhishek G R	2487232
3.	Alaparthi Sai Madhavi	2489588
4.	Byrraju Bala Durga Kiran Raju	2491107
5.	Deepika Jaibeeb Sivaraj	2488331
6.	Kajal Ashok Gaikwad	2488299
7.	Khdija Khan	2489600
8.	Nilesh Dharmraj Tale	2487449
9.	Sahil Khan	2484251
10.	Tupili Sai	2490447

ABSTRACT

The purpose of E-Medicare System is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

E-Medicare System, as described above, can lead to error free, secure, reliable and fast management system.

TABLE OF CONTENTS

1.INTRODUCTION

The "E-Medicare System" has been developed to override the problems prevailing in the practicing manual system. This software is supposed to eliminate and reduce the hardships faced by the existing system. E-Medicare System can lead to error free, secure, reliable and fast management system.

"E-Medicare System" - web application [J2EE Batches - Web Application], where users can register, login, purchase medicines e.g. Antibiotics, Antipyretics. And analgesics manage their orders in the system.

1.1 Scope and Overview:

The scope of the "E-Medicare System" will be to provide the functionality as described below. The system will be developed on a Windows operating system using JSON.

2.FUNCTIONALITIES PROVIDED BY E-MEDICARE SYSTEM ARE AS FOLLOWS:

- Provides the searching facilities based on various factors. Such as Customer, Medicine, Order.
- E-Medicare System also manage the Payment details online for Order details, Customer.
- It tracks all the information of Medicine Company, Payment etc.
- Manage the information medicine company.
- Shows the information and description of the customer and medicine.
- Editing adding and updating of records is improved which results in proper resource management of customer data.

2.1 MODULES OF E-MEDICARE SYSTEM:

- Customer Module: Used for managing the Customer details.
- Order Module: Used for managing the details of Order.
- Cart Module: Used for adding medicines.
- Medicine Module: Used for managing the Medicine details.
- Login Module: Used for managing the login details.
- Users Module: Used for managing the users of the system.
- Admin Module: Used for managing medicine details and user information.

3.SYSTEM OVERVIEW

The "E-Medicare System" should support basic functionalities (explained in section 2.1) for all below listed users.

- Admin (A)
- User (C)

3.1 AUTHENTICATION & AUTHORIZATION

3.1.1 Authentication:

Any end-user should be authenticated using a unique user-id and password.

3.1.2 Authorization:

The operations supported and allowed would be based on the user type. For example, Administrator has the rights to add product information and view customer details. He can also view order details and purchase details of a product. Whereas Customer/Buyer has a right to Add, Remove and Clear all the products from cart.

3.2 ENVIRONMENT:

The system will be developed on any Windows OS machine using JSON.

- Intel hardware machine (PC i3-2.26 GHz, 8 GB RAM, 1 TB HDD)
- Server – JSON
- Database – JSON
- Node Version 10
- Angular CLI

4.SUB-SYSTEM DETAILS

The E-Medicare System is defined, wherein all users need to login successfully before performing any of their respective operations.

Find below (section 3.1 & 3.2) tables that provides functionality descriptions for each type of user / sub-system. Against each requirement, indicative data is listed in column 'Data to include'. Further, suggested to add/modify more details wherever required with an approval from customer/faculty.

4.1 ADMIN

The administrator as a user is defined to perform below listed operations after successful login.

Objects	Operations	Data to include
Medicines	Add View Delete Modify	Medicine Id, Medicine Name, Price, description, manufacture date, expiry date and status
User	View	Username, contact no, dob, email, gender
Order and Purchase	View	User Id, item Id, Price, Quantity

4.2 USER

The user is defined to perform below listed operations after successful login.

Objects	Operations	Data to include
User	Register	User Id, Username, Password, Email, Phone, Number, dob, gender
Product	Add to Cart Delete from Cart Delete all Products from cart	Product Id, Product Name, Price, Quantity, Status
Checkout	Add User Details and Price	Cart-Id, User-Id and Total Price

4.3 LOGIN | LOGOUT
[Web Application - JSON]

- Go to Registration screen when you click on Register link.
- Go to Success screen when you login successfully after entering valid username & password fetched from the database.
- Redirect back to same login screen if username & password are not matching.
- Implement Session tracking for all logged in users before allowing access to application features. Anonymous users should be checked, unless explicitly mentioned.

5.DATA ORGANIZATION

This section explains the data storage requirements of the Product Order Entry System and **indicative** data description along with suggested table (database) structure. The following section explains few of the tables (fields) with description. However, in similar approach need to be considered for all other tables.

5.1 TABLE: ADMIN_DETAILS

The admin specific details such as user-id and password. This table contains information related to Admin details.

Field Name	Description
User id	User-ID is auto generated after registration and it is used as Login-ID.
Password	Admin Password

5.2 TABLE: USER_SIGNUP_DETAILS

The user specific details such as user-id, first name, last name, date of birth, email.

Field Name	Description
User Id	User Id is the primary field, it is auto generated after registration and it is used as Login-Id
Name	First name of the User/Customer
Email	User/Customer Email Id
Contact No	10-digit contact number of the User/Customer
Password	Password of the User/Customer

5.3 TABLE: USER_LOGIN_DETAILS

This table contains information related to User Login details.

Field Name	Description
User id	User-Id of the Customer. It is auto generated after registration
Password	Password of the Customer/User

5.4 TABLE: MEDICINE_DETAILS

This table contains information related to Medicine details.

Field Name	Description
Med id	Here Medicine Id will be Primary Key
Med Name	Name of the Medicine e.g., Aspirin, Paracetamol etc.
Description	Description of the Medicine
Price	Price of the Medicine
Status	Medicine Availability Status, e.g., Pending or Placed

5.5 TABLE: CART_DETAILS

This table contains information related to cart details.

Field Name	Description
S.No	Describe the serial of product.
Medicine Name	Describe the name of medicine.
Product Image	Image of Medicine.
Description	Description of Medicine.
Price	Price of the particular Medicine.
Quantity	Quantity of the Product e.g., 2 or 3 no of Products.
Total	Total price of Medicine.
Action	Particular has to be deleted.

6. REST APIs TO BE BUILT.

- Create following REST resources which are required in the application

1.Creating User Entity: Create JSON SERVER.

Technology stack:

- **JSON Server** : JSON Server is a Node Module that you can use to create demo REST JSON services within a short span of minutes.

- A common use of JSON is to exchange data to/from a web server. When receiving data from a web server, the data is always a string. Parse the data with JSON.

- A JSON document database is a type of non-relational database that is designed to store and query data as JSON documents, rather than normalizing data across multiple tables, each with a unique and fixed structure, as in a relational database.

Here will have multiple layers into the application:

1.Create an Entity: User

2.Create a User signup Repository interface and will make use of JSON Server

a) Will have find By User Name method.

b) Add the User details

3.Create a User signup Service class and will expose all these services.

4.Finally, create a User Rest Controller will have the following Uri's:

URI	METHODS	Description	Format
/users	GET	This gets function retrieves all users in the database.	JSON
/users/userid	GET	This function retrieves the data based on the userid.	JSON
/users	POST	Add the user details	JSON
/users/userid	PUT	It modifies the existing user details	JSON
/users/userid	DELETE	It Delete user by id	String

2. Creating Medicine Entity:

Build a RESTful resource for Medicine manipulations, where CRUD operations to be carried out. Here will have multiple layers into the application:

1. Create an Entity: Medicine
2. Create a MedicineRepository interface and will make use of JSON Database.
   - Will have findByMedicineName method.
   - Add the Medicine details method.
   - Will have deleteMedicineById method.
   - Will have findAllMedicine method.
3. Create a MedicineService class and will expose all these services.
4. Finally, create a MedicineRestController will have the following Uri's:

URI	METHOD	Description	Format
/Medicines	GET	Get all available medicines	JSON
/medicine/id	GET	Get medicine description based on the given id	JSON
/medicine	POST	Add the medicine details	JSON
/medicine/id	DELETE	Delete a medicine based on medicine id	JSON

3.Creating Cart Entity:

Build a RESTful resource for **Cart** manipulations, where following operations to be carried out. Here will have multiple layers into the application:

1. Create an Entity: Cart
2. Create a CartRepository interface and will make use of JSON Database
   - Add the Cart details
   - Will have deleteCartById method to remove item with specific product Id from cart.

- Will have All cart details.

3. Create a CartService class and will expose all these services.

4. Finally, create a CartRestController will have the following Uri's:

URI	METHOD	Description	Format
/Cart	GET	It gets available cart details	JSON
/Cart	POST	Add the medicines to cart	JSON
/Cart/id	PUT	Update the cart details	JSON
/Cart/id	DELETE	Delete the medicines from cart	JSON

4. Creating order Entity:

Build a RESTful resource for order manipulations, operations to be carried out. Here will have multiple layers into the application:

1. Create an Entity: Vieworder

2. Create a OrderRepository interface and will make use of JSON Database.

- Will have findByorderName method
- Add the order details method
- Will have deleteorderById method
- Will have findAllorders method

3. Create a VieworderService class and will expose all these services.

4. Finally, create a MedicineRestController will have the following Uri's:

URI	METHODS	Description	Format
/orders	GET	Get all orders.	JSON
/order/id	GET	Get orderdescription based in the given id.	JSON
/order	POST	Add the order details	JSON
/order/id	POST	Delete order based on medicine id	JSON

7.FEASIBILITY STUDY:

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements

A. Economic Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor.

• All hardware and software cost has to be borne by the organization.

• we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system

B. Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system.

C. Operational Feasibility

This Project is very user friendly and all inputs to be taken all self-explanatory even to a layman.

8. ASSUMPTIONS

Project assumption is the events or conditions that are most likely to occur when a project life-cycle takes place. It is also deemed as an element in the planning phase of a particular project that is assumed to be considered as

true and actual. Assumptions are beliefs based on previous experience and the information available to you.

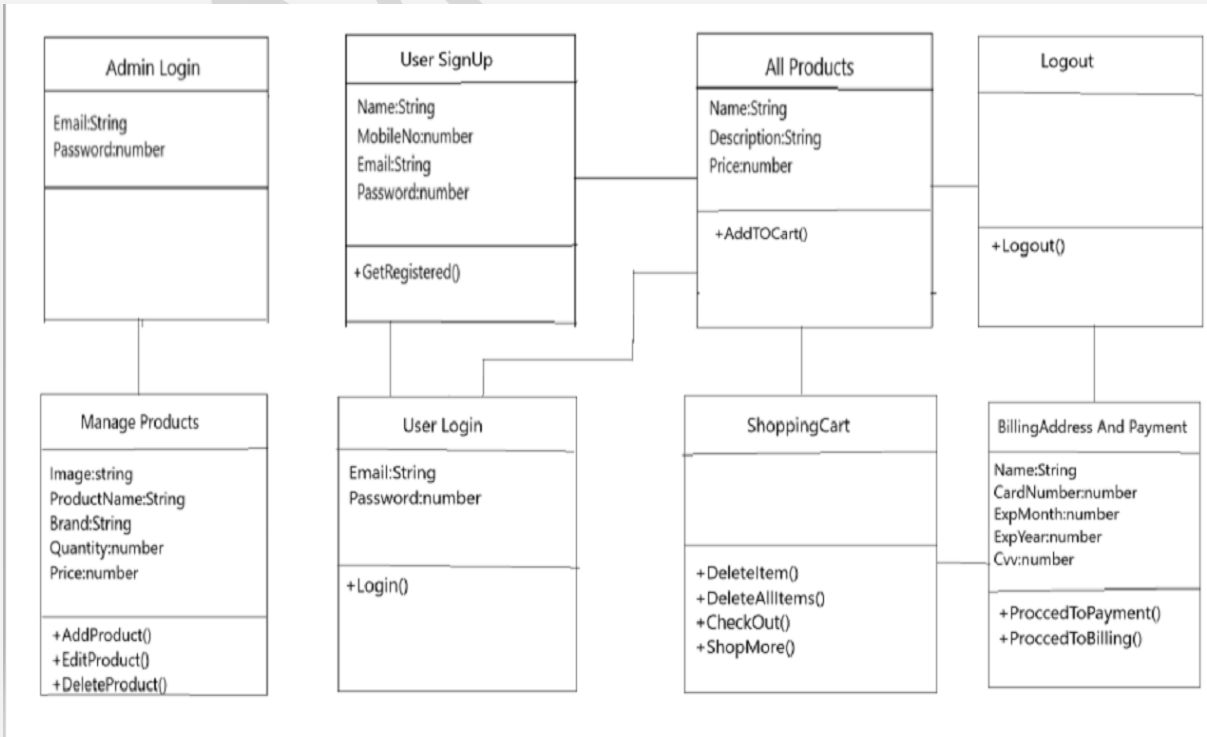
- The product does require back-end database server MySQL for storing for storing the username and password for different types of users of the system.
- User must be trained for basic computer functionalities.
- User must have the basic knowledge of English.
- The system must be able to respond to database software within reasonable time
- User Interface: The type of client interface (front-end) to be supported - Angular based
- The administrator can add and remove medicine products into the database on a weekly basis.
- You must not allow user to add same medicine product twice.
- When you add medicine product into cart the No. of medicine Products selected will be incremented.
- If you delete the medicine product from the cart, the counter will be decremented.
- The clear will remove all the medicine products so that the No. of medicine products will be zero
- The total amount will be calculated based on the No. of medicine product, accordingly, change the product counter & total amount.

9.UML DIAGRAMS

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

9.1 CLASS DIAGRAM:

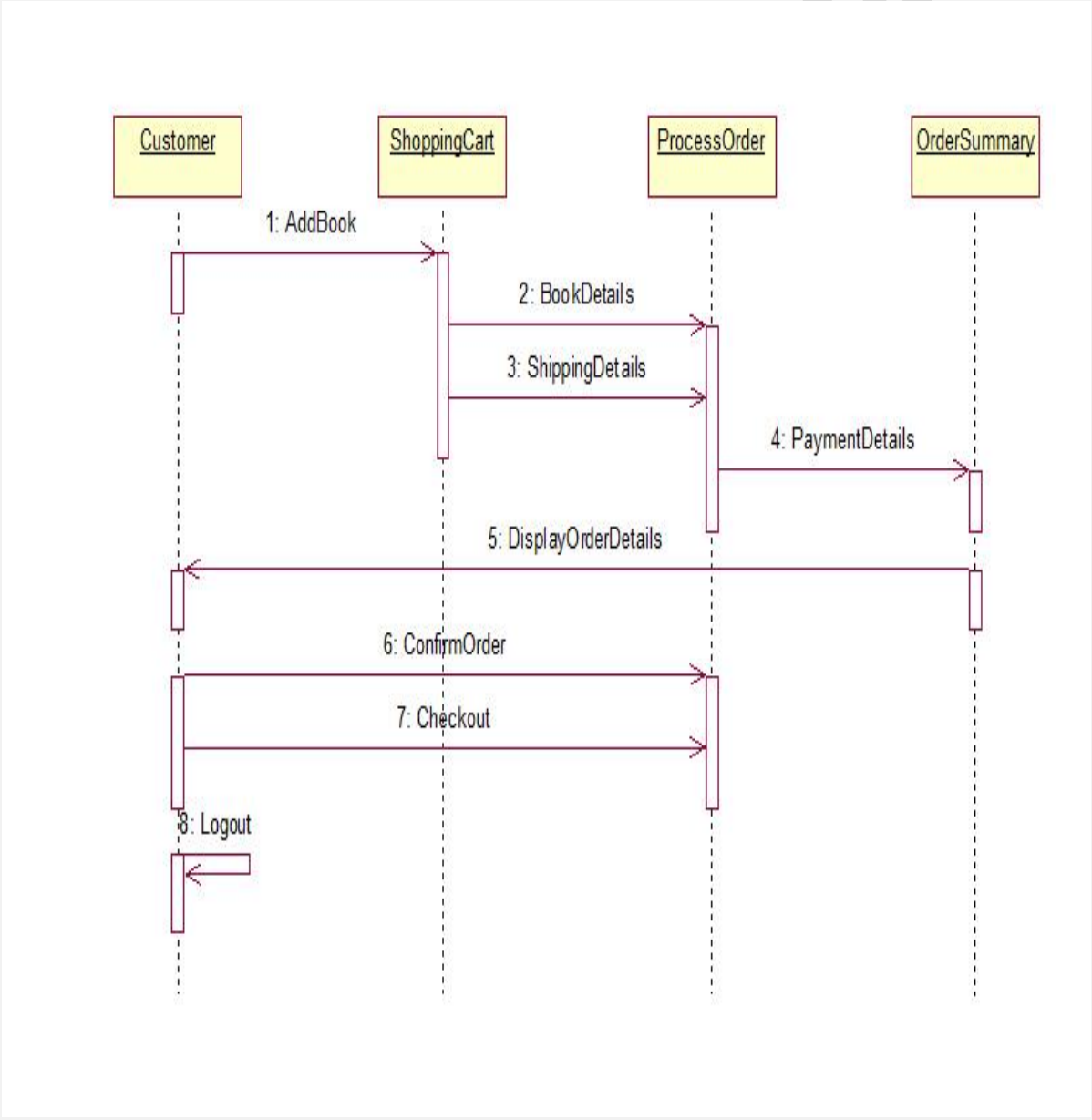
- Admin: Mainly responsible for account management and adding or modifying new product categories.
- Customer: Members can perform all the activities that guests can, in addition to which, they can place orders and add new products to sell.
- Product: This class will encapsulate the entity that the users of our system will be buying and selling. Each Product will belong to a Product Category.
- Guest: Guests can search for and view products, and add them in the shopping cart. To place an order, they have to become a registered member.
- Cart: Users will add product items that they intend to buy to the shopping cart.
- Payment: This class will encapsulate the payment for an order. Members can pay through credit card or electronic bank transfer.



9.2 SEQUENCE DIAGRAM:

A sequence diagram is structured in such a way that it represents a timeline which begins at the top and descends gradually to mark the sequence of interactions. Each object has a column and the messages exchanged between them are represented by arrows.

The diagram presented here will show you a detailed illustration of the sequence of events happen in E-Medicare System.



10.TESTING:

1.Unit Testing:

Each module is considered independently. it focuses on each unit of software as implemented in the source code it is white box testing.

2.Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules.

3.Validation Testing:

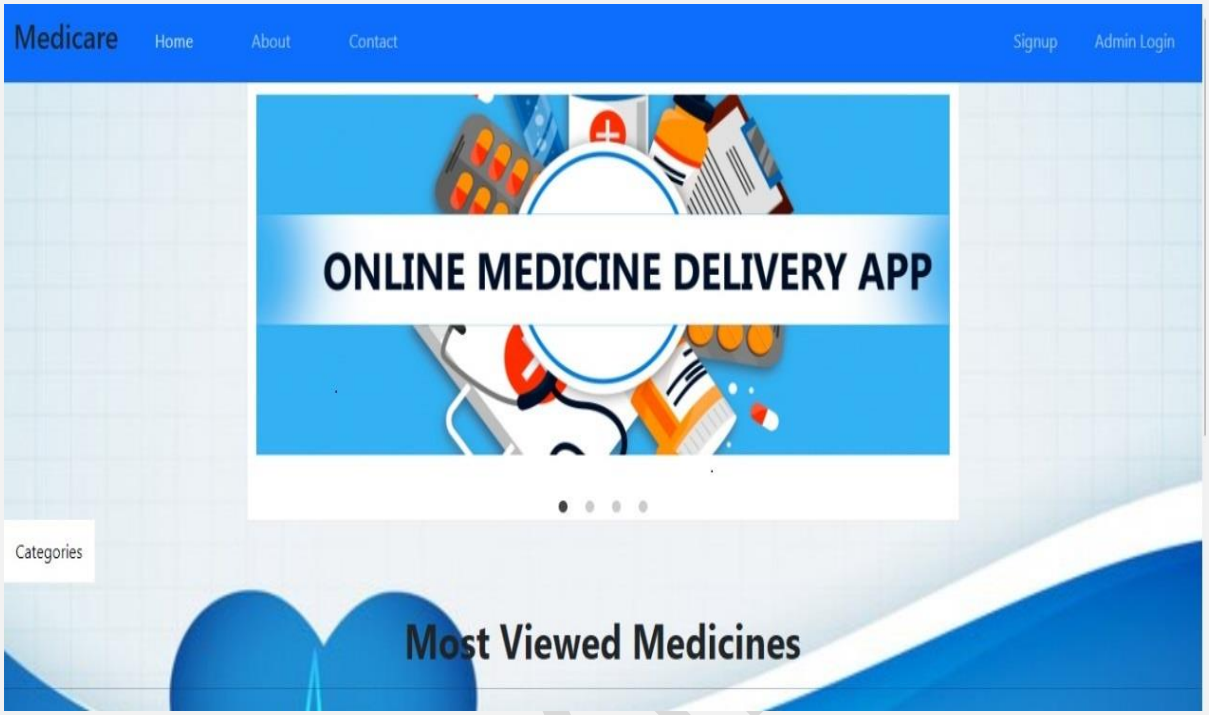
Validation testing was performed to ensure that all the functional and performance requirements are met.

4.System Testing:

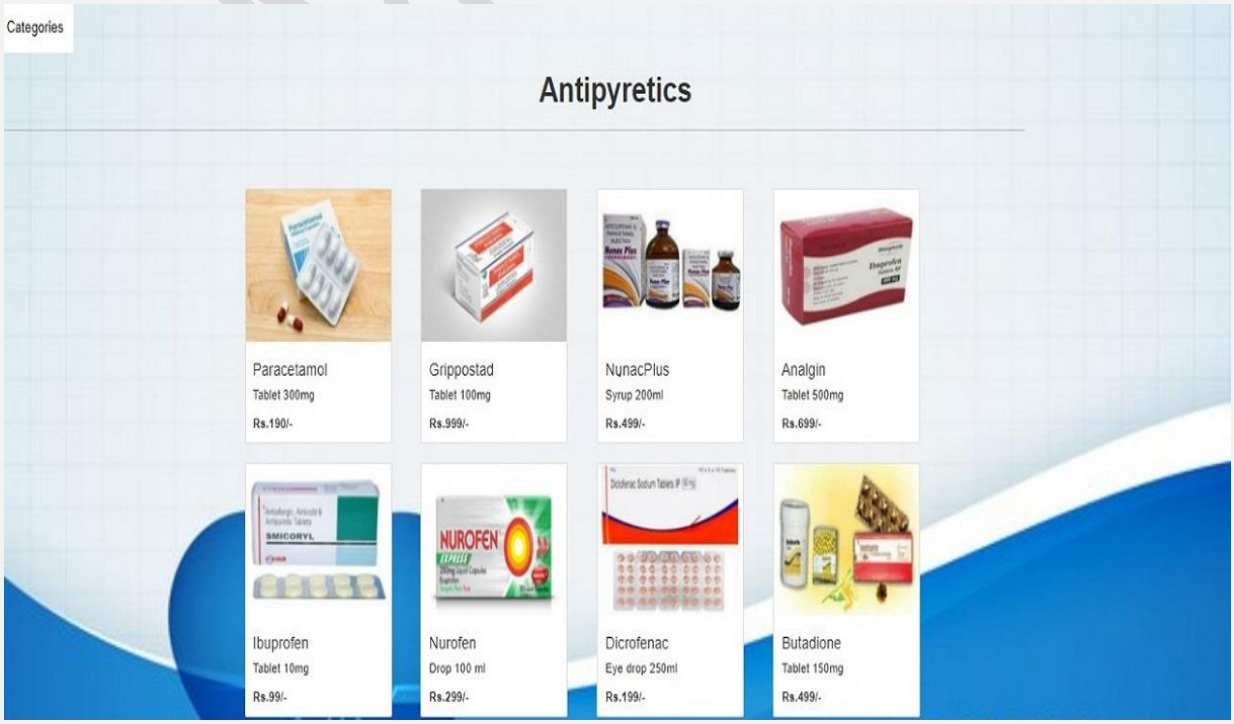
It is executing programs to check logical changes made in it with intention of finding errors, a system is tested for online response, volume of transaction, recovery from failure etc.

11.Output Screenshots:-

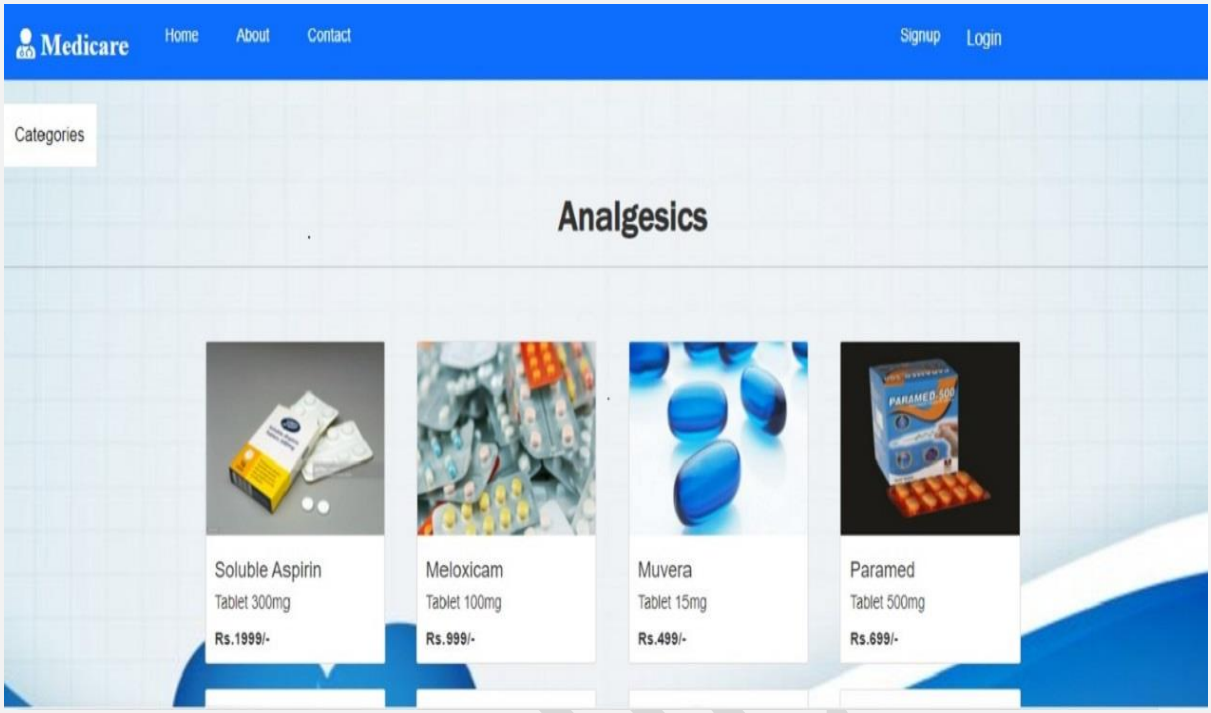
Homepage:



Categories: 1-Antipyretics:



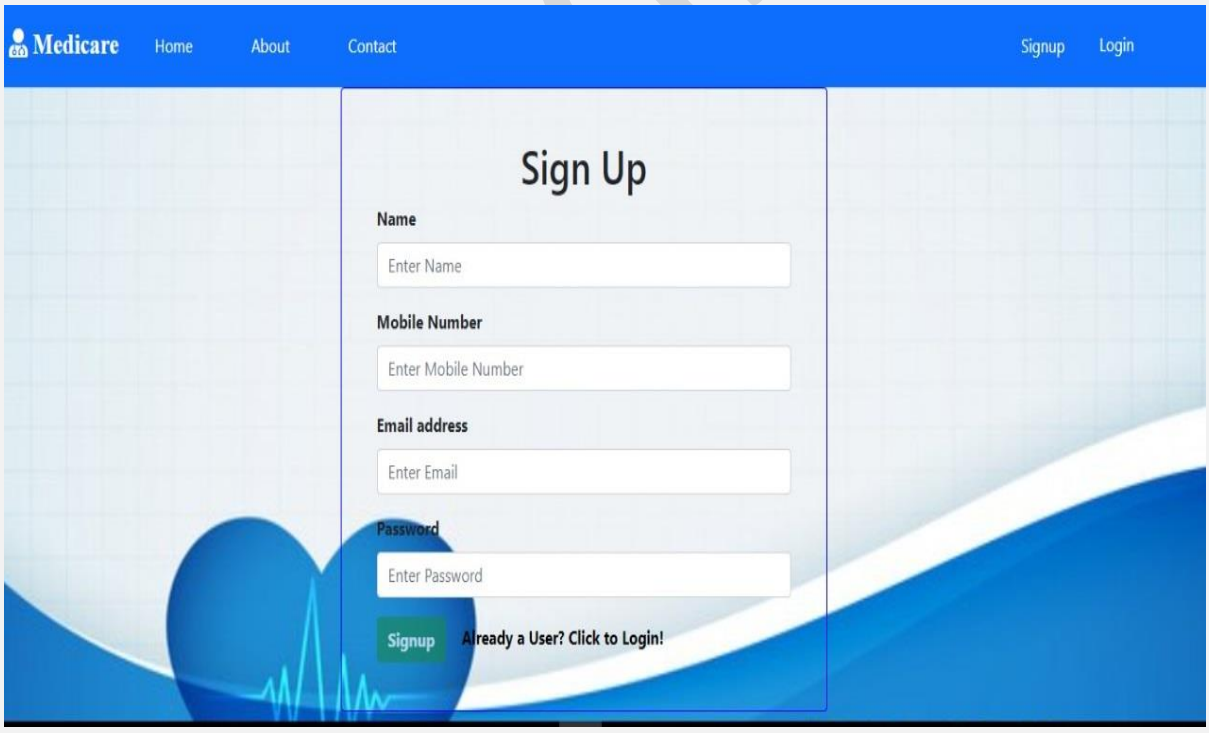
2-Analgesics:



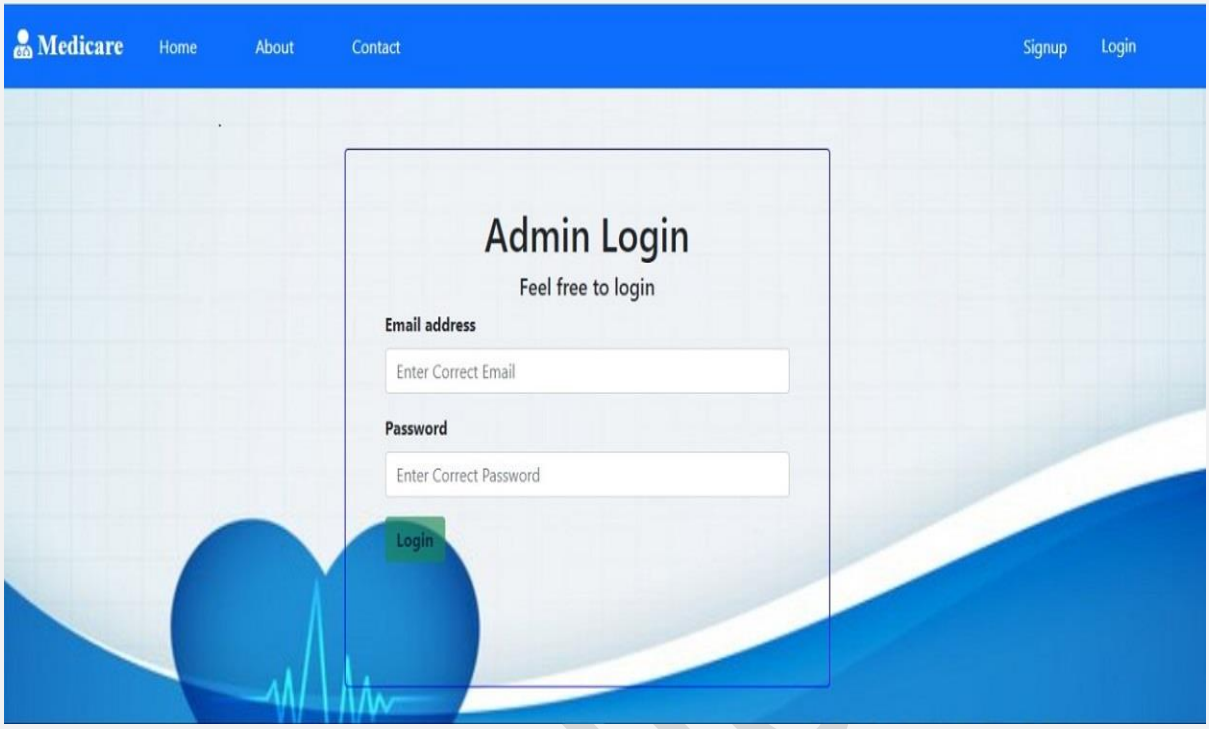
3-Antibiotics:

About:

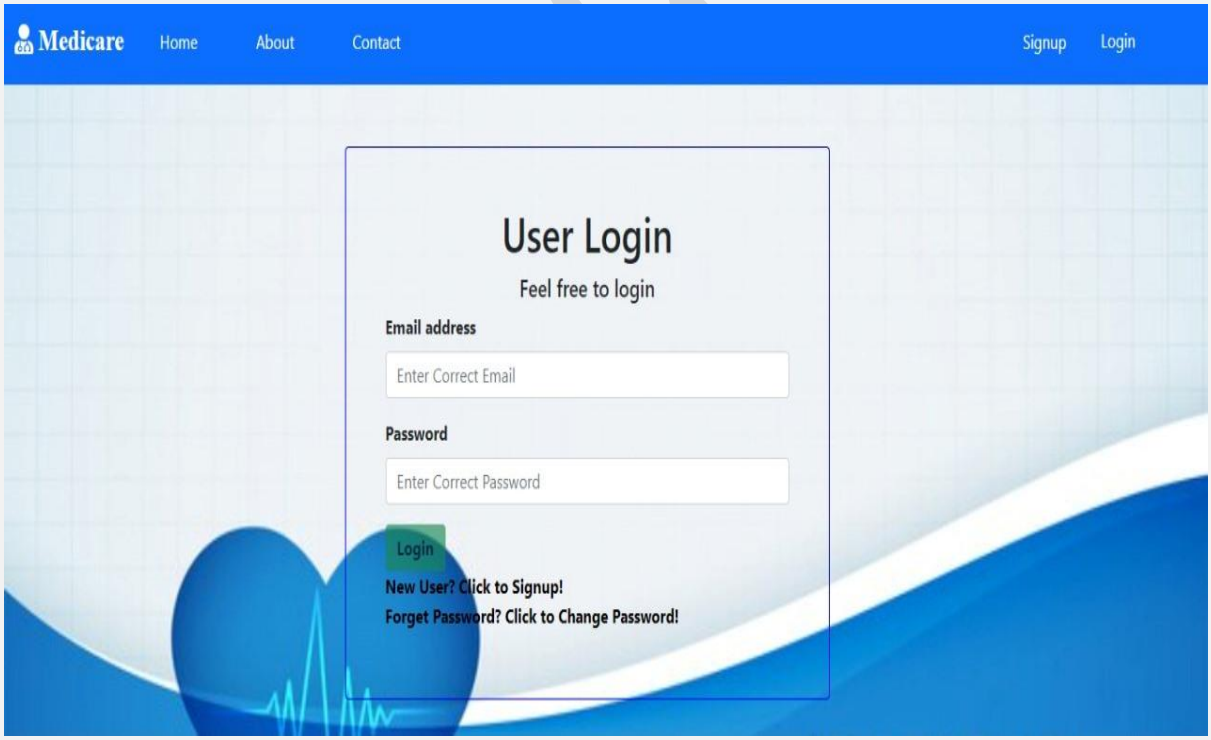
Sign Up:



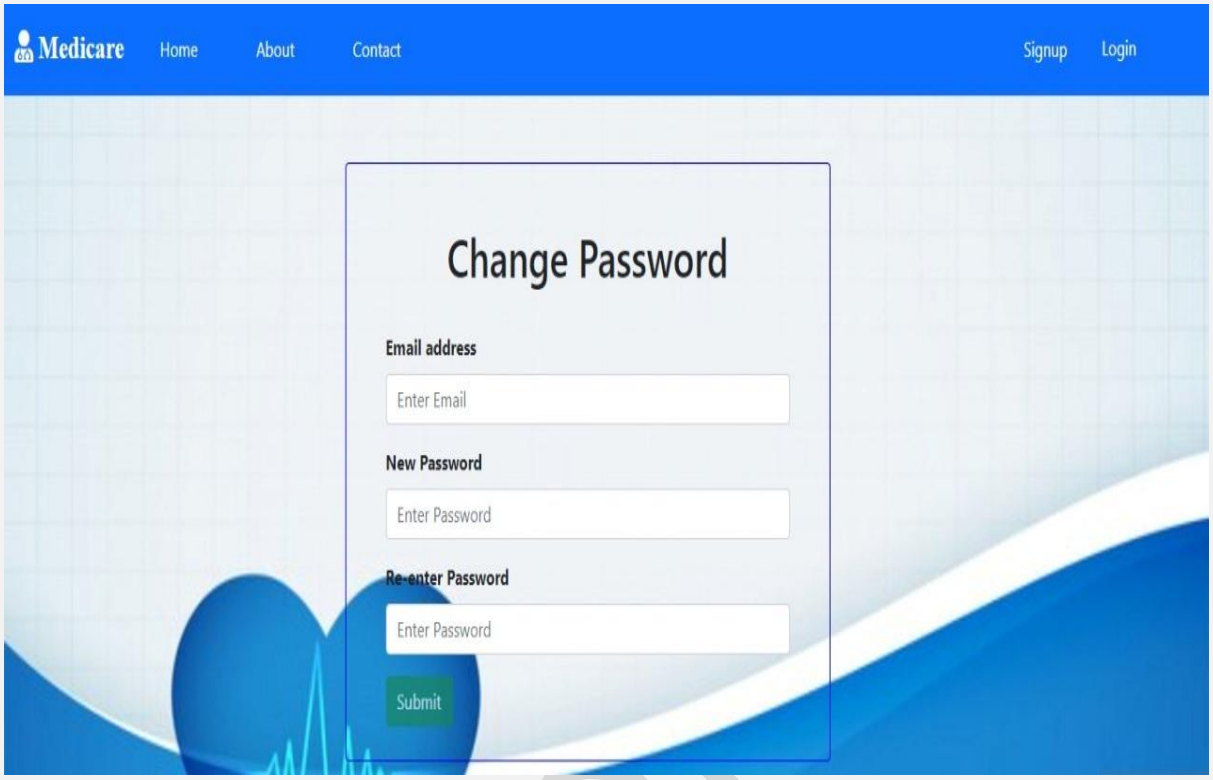
Admin Login:



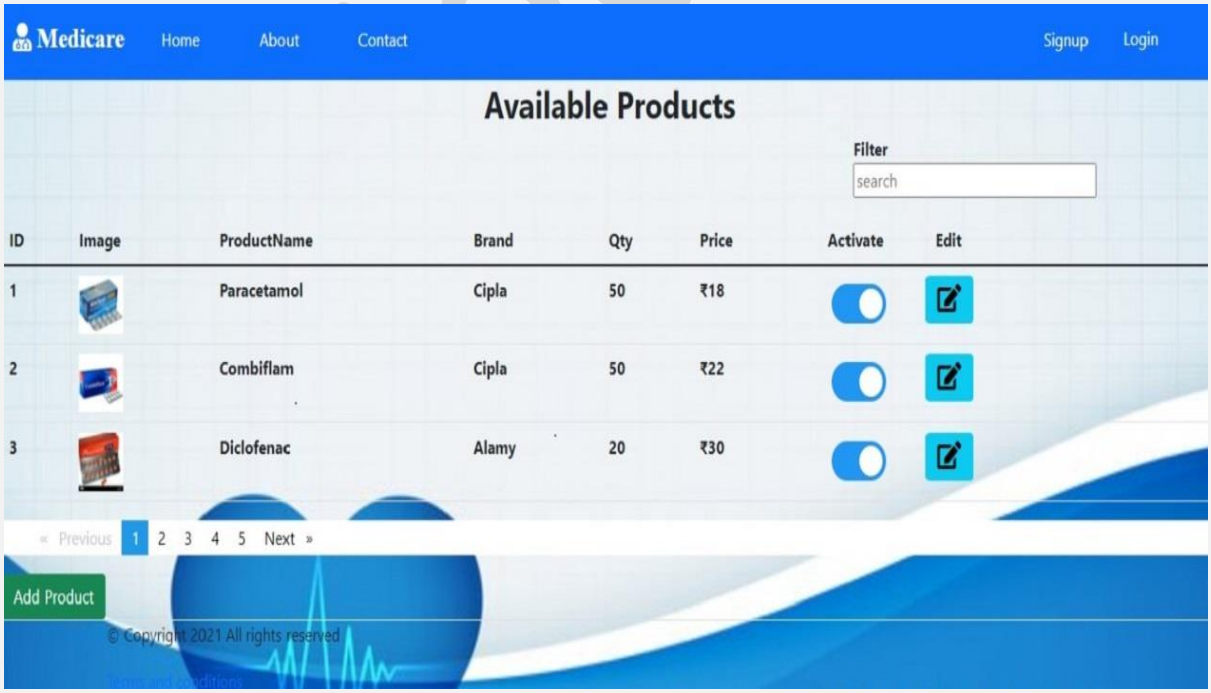
User Login:



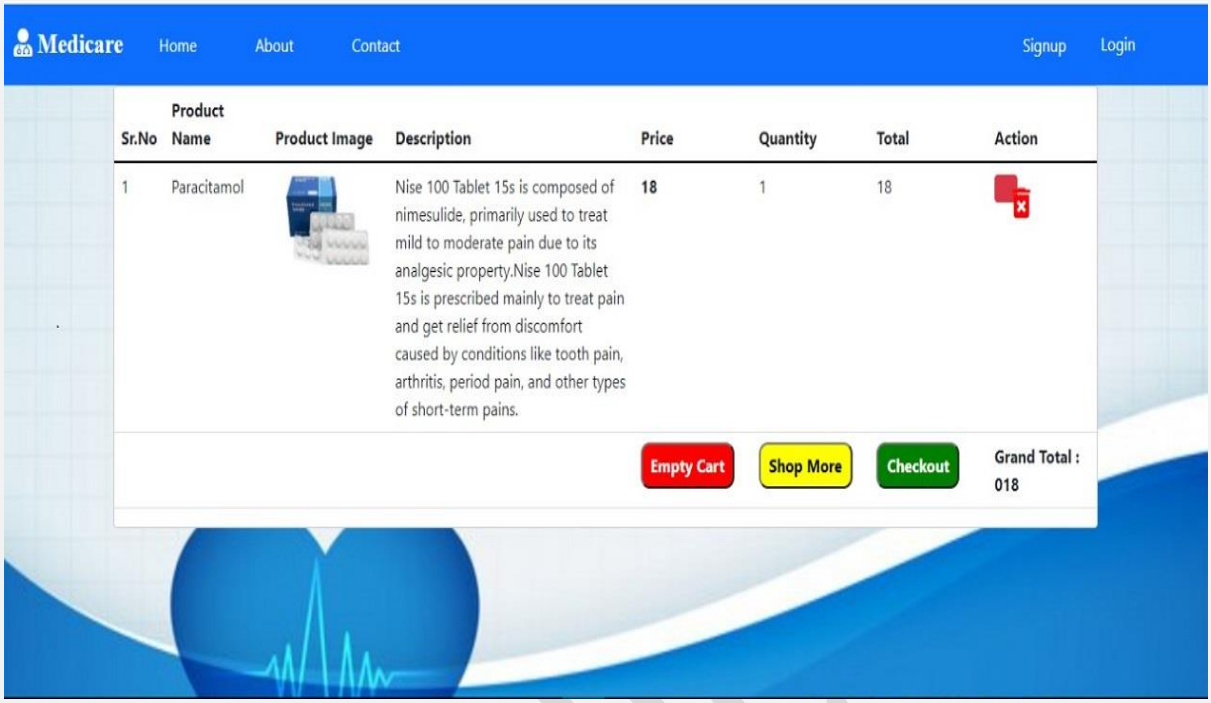
Forgot Password:



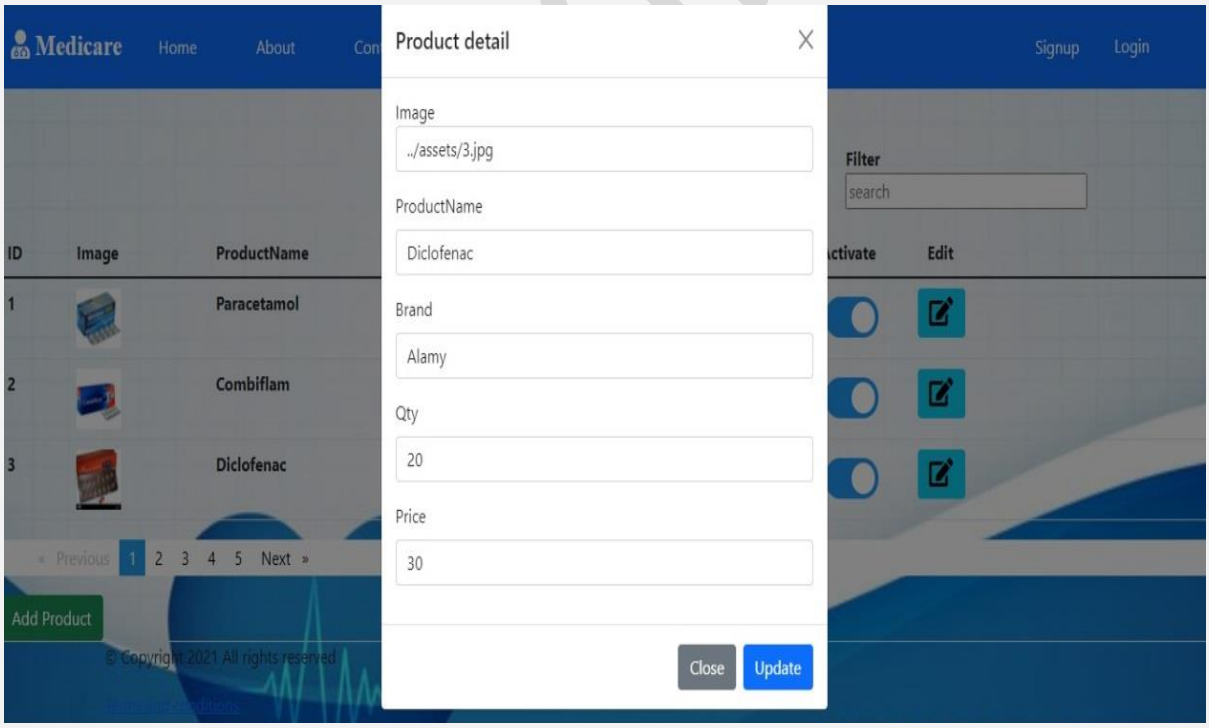
Available Products:



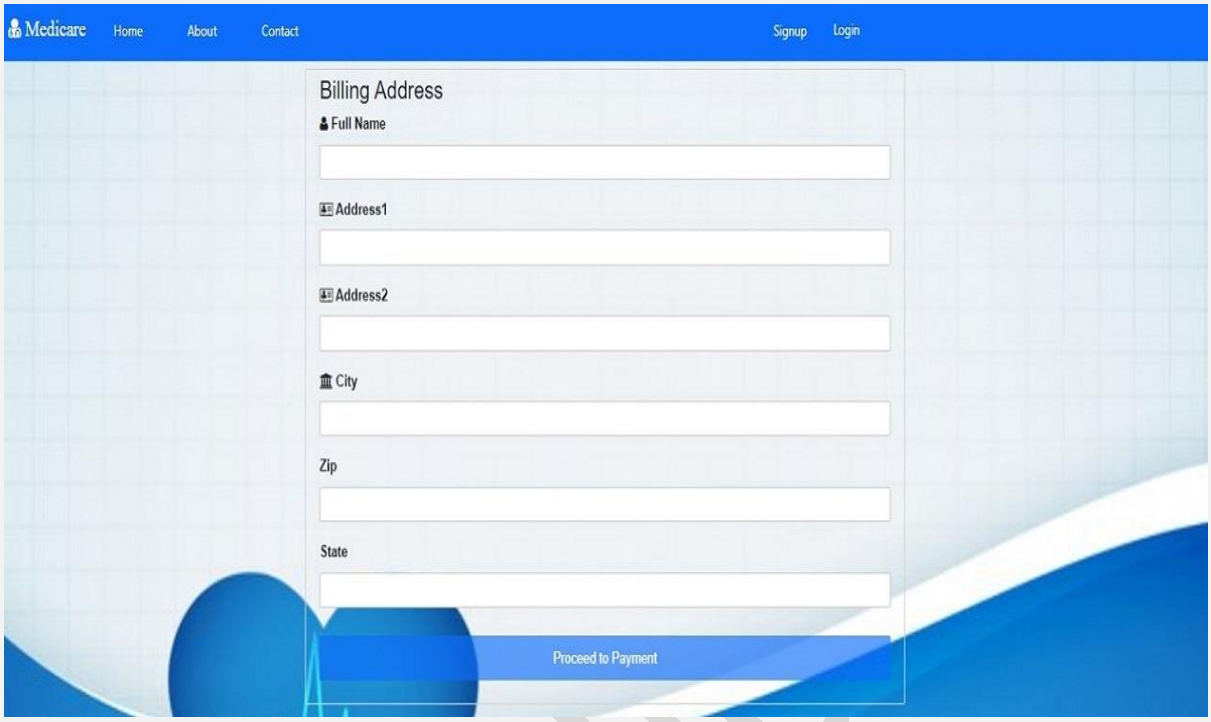
Cart List:



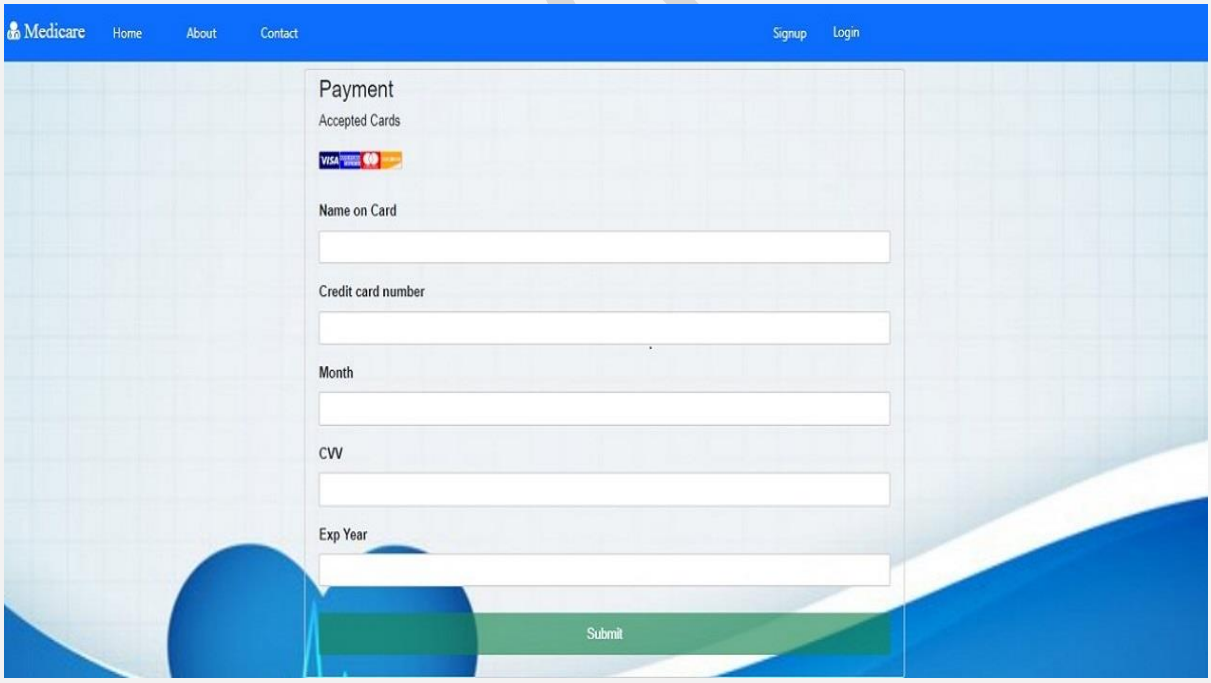
Update Medicine Details:



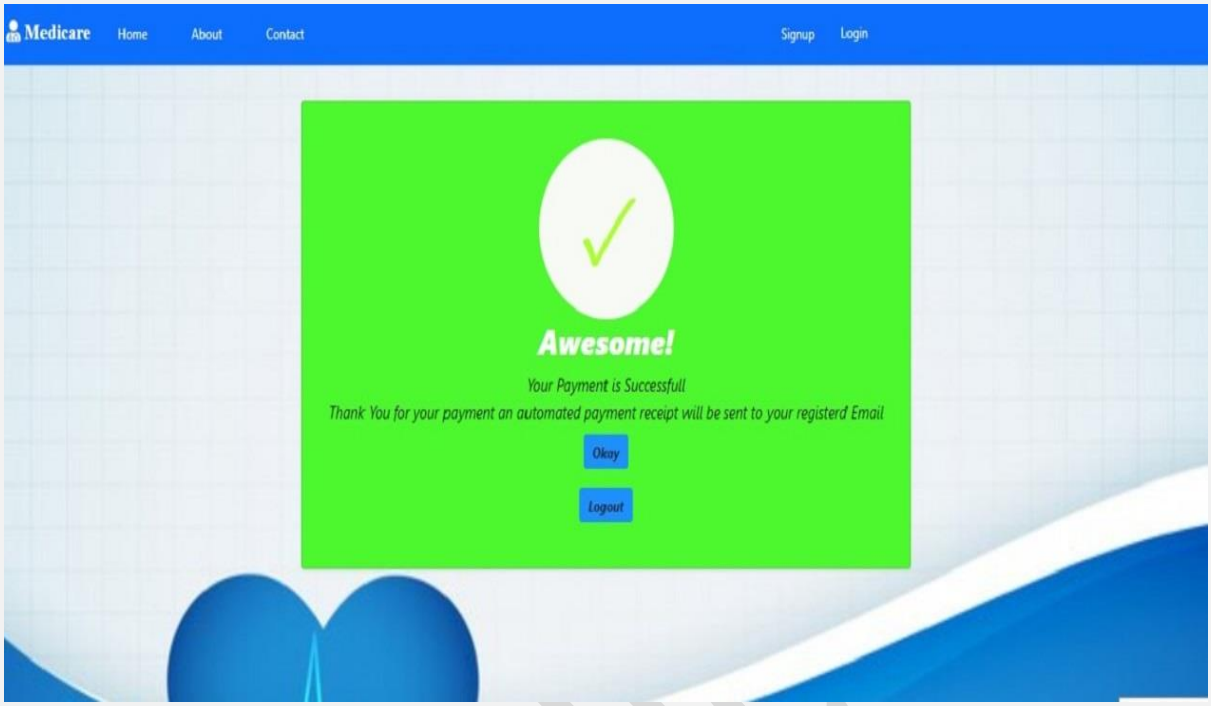
Billing Address:



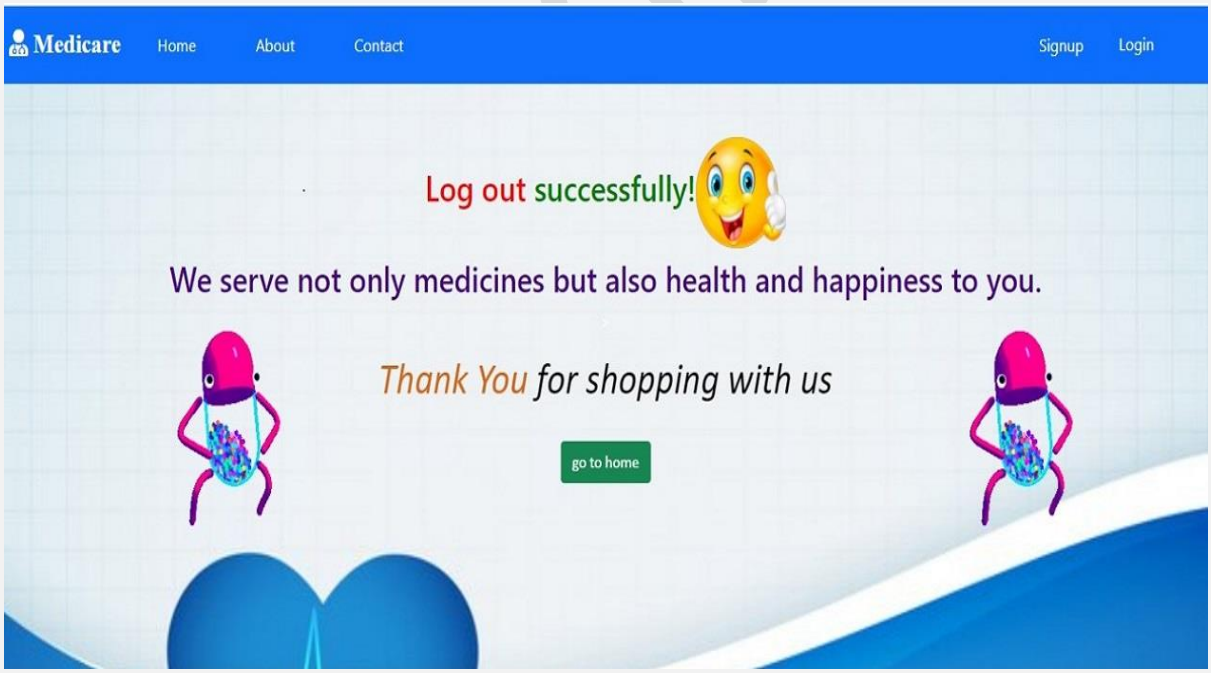
Payment Page:



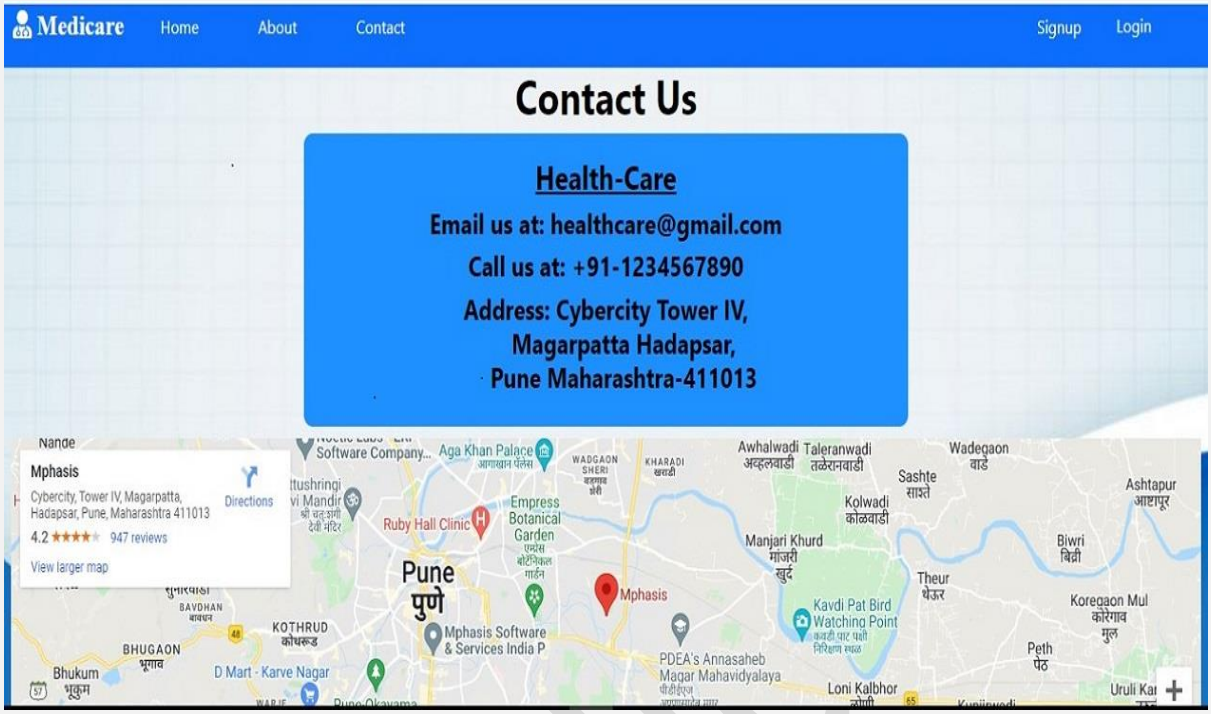
Payment Successful Page:



Logout:



Contact Us:



12.1 Advantages:

- ✓ User can view details of the medicines without going anywhere.
- ✓ User can change their delivery address
- ✓ It is convenient for users as this system provides accurate cost and description of the system.
- ✓ The website is flexible to be used and for e-shopping.
- ✓ User can view different categories of product of different pharma company at a single place.

12.2 Future Scope:

- ✓ Use of digital technologies and telecommunications, such as computers, the Internet, and mobile devices, to facilitate health improvement and health care services.

- ✓ In future, with this we can implement to take appointment with the doctor And can connect to any doctor in the world.

- ✓ Assist individuals in monitoring their own health conditions, such as heart disease, diabetes, pregnancy, mental health, and more.

- ✓ The purpose of all these works aims to provide user a pleasuring experience and great user interaction. Technology changes the life, so

- ✓ Learning new technique constantly is quite necessary and important.

12.3 References By:

1. **W3Schools:-** **(https://www.w3schools.com/)**
2. **Bootstrap:-** **(https://getbootstrap.com/)**
3. **GeeksForGeeks:-** **(https://practice.geeksforgeeks.org/courses)**
4. **YouTube:-** **(https://www.youtube.com/)**