ONLINE MEDICAL STORE [MEDICURE]

A Major project submitted for the partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATION

Submitted By

Tanmay Samanta

Roll No : 18871022060 Registration No: 221880510058 of 2022-23 Session : 2022-2024

Under The Supervision Of

Mr. Subhendu Saha
Asst. Professor
Department Of MCA
Techno College Hooghly

Techno College Hooghly
Dharampur, Shantiniketan, Near Khadina more
G.T. Road, Chinsurah, Hooghly-712101
Phone: 033-2680-2389 / 6565
Website: www.technoindiahooghly.org Email: info@technoindiahooghly.org

Techno College Hooghly
Dharampur, Shantiniketan, Near Khadinamore
G. T. Road, Chinsurah, Hooghly - 712101
Phone: (033) 2680-2389
E-mail: info@technocollegehooghly.org
Website: www.technocollegehooghly.org

Certificate

This is to certify that the project entitled ***"Online Medical Store (Medicure)"*** *submitted by* ***Tanmay Samanta****, University Roll No:* ***18871022060*** *as Assigned Project (major) for the partial fulfillment of degree of Master of Computer Application is worth of acceptance.*

(Mr. Subhendu Saha)	**(Mr. Subrata Saha)**	**(Dr. Pintu Pal)**
Project Supervisor	Head of the Department, MCA	Principal
	Techno College Hooghly	Techno College Hooghly

ACKNOWLEDGEMENT

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behaviour, and acts during the course of study.

We express sincere gratitude to **Dr. Pintu Pal** worthy **Principal** for providing us an opportunity to undergo this project at **TECHNO COLLEGE HOOGHLY.**

We are also thankful to Mr. Subhendu Saha and our HOD Mr. Subrata Saha for their support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

Lastly, we would like to thank the almighty and our parents for their moral support and our friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

Tanmay Samanta

_____

TABLE OF CONTENTS

PROJECT ABSTRACT:

An **online pharmacy** is a pharmacy that operates over the Internet and sends orders to customers through mail, shipping companies. A web-based pharmacy is an online system that provides a platform for customers to purchase medicinal drugs and E-services online, allowing the customer to receive wide range of medicines/services from the comfort of their homes within a short time.

Our web application helps user view details of the medicines without going anywhere.

INTRODUCTION

A web-based pharmacy is an online system that provides a platform for customers to purchase medicinal drugs and E-services online, allowing the customer to receive medicines/services in the comfort of their homes within a short time. 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.

OBJECTIVE AND SCOPE

The project is on developing an online medical shop website. The objectives of the website are to provide an easy-to-use interface for customers to search for medicines, view product details, and order products for home delivery. It also allows customers to query about medicines. It also has following objectives: To provide searching facilities on the basis of various factors like price. To manage the user details for analyzing sales and expenses. To keep track and manage all information.

Scope of the project is very broad in terms of other systems. Few of them are: -

• Provides the searching facilities based various factors, such as category, price, availability.

• It stores all the details of the medicines, so the customers can take an informed decision.

• Manage the orders directly by the customers.

• To increase efficiency of medicine warehouse.

• Manage the information of the user.

THEORETICAL BACKGROUND

What is React?

React is a JavaScript library developed by Facebook, which among other thing, was used to build instagram.com. Its aim is to allow developers to easily create fast user interfaces for websites and applications alike. The main concept of React.js is virtual DOM.

Why use React?

- **Easy to Learn:**

Unlike Angular and Vue, learning React is like five-finger exercise, well of course on your keyboard! No wonder why, that's a focal reason for React's rapid adoption. It allows businesses to complete projects more quickly. React is more credible to be used by large firms as it is such a framework that is easy to get going with.

- **Reusable Components:**

React JS provides developers with reusable components that they can use to create new applications. Reusability is like a developer's miracle cure. This platform enables developers to reuse any react component created for another application that performs the same function. As a result, development effort is reduced while flawless performance is maintained. One of the chief benefits of using React JS is that developers can reuse components as many times as they need to. Furthermore, changes made in one section of the application have no effect on other sections of the application. If you thought React was only for web development, you were dead wrong! Facebook has already developed the React Native framework for creating mobile apps for both the Android and iOS platforms.

- **Fast Rendering:**

When developing a complex, high-load app, it is essential to define the structure of the app at the beginning as it can greatly influence the app's performance. Simply put, as the DOM is tree-structured, even little changes at the root can cause major impacts on others. To solve this problem, Facebook has brought up a virtual DOM feature. Virtual DOM, which allows all modifications to be tested first for calculating risks. This approach contributes to app performance and a better user experience.

- **SEO Friendly:**

Another benefit of React js is that it can cope with a common search engine failure to comprehend JavaScript-heavy apps. React significantly reduces load time of websites, which helps businesses to lead on the top spot-on Google's Search Engine Result Page.

There are some of the frameworks that React possesses which improves SEO, one of them is NextJs. The rendering part of NextJs is moved to the server, removing the need for the client to process the data. Since the server pre-renders the pages before sending the final HTML to the client, there is less JS to load, which improves performance and SEO.

- **Strong Community Support:**

One of the main reasons to choose React for your project is a strong community behind it. A massive range of individual React developers contribute to creating a more robust frontend framework on a daily basis.

Proficient Data Binding:

ReactJS lags behind one-way data binding. So that anyone can track all the changes made on particular portions of the data. This is often an illustration of its simplicity.

The Virtual DOM:

The virtual DOM(Document Object Model) is one of the most intriguing aspects of React. While creating an app with a lot of user interaction and data changes, you should think about how the structure of the app will affect speed. Extensive DOM manipulation can cause congestion in performance. Worse, as the DOM is tree-structured, even little changes at the root can cause major impacts on others.

A virtual DOM is used by React to solve this problem. This is a virtual version of the DOM, as the name implies. Any new view updates are made on the virtual DOM first, which is stored in memory rather than on your screen. It then finds the most efficient method of making these modifications, and only those changes are applied to the real DOM. This ensures that the real DOM is updated as quickly as possible, resulting in improved performance and a cleaner user experience overall.

Great Developer Tools:

React is a vast library for creating user interfaces; additionally it is supported by React Developer Tools which are browser extensions available for both Google Chrome and Mozilla Firefox.

Why use Node?

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.

This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks. Almost no function in Node.js directly performs I/O, so the process never blocks except when the I/O is performed using synchronous methods of Node.js standard library. Because nothing blocks, scalable systems are very reasonable to develop in Node.js. Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. In Node.js, there is no such start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user. HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment. Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

What is MongoDB?

MongoDB is an open-source document database built on a horizontal scale-out architecture that uses a flexible schema for storing data. Founded in 2007, MongoDB has a worldwide following in the developer community.

Instead of storing data in tables of rows or columns like SQL databases, each record in a MongoDB database is a document described in BSON, a binary representation of the data. Applications can then retrieve this information in a JSON format. Document databases are highly flexible, allowing variations in the structure of documents and storing documents that are partially complete. One document can have others embedded in it. Fields in a document play the role of columns in a SQL database, and like columns, they can be indexed to increase search performance.

From its founding, MongoDB was built on a scale-out architecture, a structure that allows many small machines to work together to create fast systems and handle huge amounts of data.

MongoDB has always focused on providing developers with an excellent user experience, which, in addition to all its other properties, has made MongoDB a favorite of developers worldwide for a wide variety of applications.

Why use MongoDB?

MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas.

As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. This format directly maps to native objects in most modern programming languages, making it a natural choice for developers, as they don’t need to think about normalizing data. MongoDB can also handle high volume and can scale both vertically or horizontally to accommodate large data loads.

MongoDB was built for people building internet and business applications who need to evolve quickly and scale elegantly. Companies and development teams of all sizes use MongoDB for a wide variety of reasons.

- Document Model

The document data model is a powerful way to store and retrieve data in any modern programming language, allowing developers to move quickly.

- Deployment Options

MongoDB is available in any major public cloud (such as AWS, Azure, and Google Cloud) through MongoDB Atlas, in large data centers through the Enterprise Advanced edition, or free through the open-source Community edition.

- Get Started Quickly

MongoDB has a great user experience for developers who can install MongoDB and start writing code immediately.

- Fully Scalable

MongoDB's horizontal, scale-out architecture can support huge volumes of both data and traffic.

- Find Community

MongoDB has developed a large and mature platform ecosystem. It has a worldwide community of developers and consultants, making it easy to get help. It also has an enterprise-grade support offering.

Using MongoDB enables your team to go further and faster when developing software applications that handle data of all sorts in a scalable way.

MongoDB is an excellent choice if you need to: Support rapid iterative development. Enable collaboration of a large number of teams. Scale to high levels of read and write traffic. Scale your data repository to a massive size. Evolve the type of deployment as the business changes. Store, manage, and search data with text, geospatial, or time-series dimensions.

MongoDB as a company has grown because the number of use cases with these characteristics continues to grow.

PROBLEM DEFINITION

- Users can add the products on the click of a button to their cart.
- Users can get the product availability on very easily on the product details page.
- Admin user can add categories to bifurcate among different types of products.
- Users can place their orders very easily and get them delivered at their doorstep.
- Users can make the payment as per their wish from the wide range of payment methods.
- Users can search from the wide range of products very easily at a single place.

SYSTEM ANALYSIS AND DESIGN VIA USER REQUIREMENT

All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document. The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

System Analysis:

The objective of the system analysis activity is to develop structured system specification for the proposed system. The structured system specification should describe what the proposed system would do; independent of the technology, which will be used to implement these requirements. The structured system specification will be called the essential model (also known as logical model).

The essential model may itself consist of multiple models, modeling different aspect of the system. The data flow diagrams may model the data and their relationships and the state transition diagram may model time dependent behavior of the system. The essential model thus consists of the following: -

- Context diagram
- Labelled data flow diagrams
- Process specification for elementary bubbles
- Data dictionary for the flow and stores on the DFDs.

System Design:

System design involves transformation of the user implementation model into software design.

The design specification of the proposed system consists of the following: -

- Database scheme
- Structure charts
- Pseudo codes for the modules in structure charts

FEASIBILITY STUDY

Why feasibility study is required?

Feasibility study is made to see if the project on completion will serve the purpose 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 area that were considered very carefully during the feasibility study of this project such as Technical, Economic, Social and Operational feasibilities.

- Technical feasibility-

The engineering feasibility of the project in viewed in the technical feasibility. Certain important engineering aspects are covered which are necessary for the designing of the project like civil, structural and other relevant aspects. Technical capability of the projected technologies and the capabilities of the personnel to be employed in the project are considered. In certain examples especially when projects are in third

world countries, technology transfer between cultures and geographical areas should be analyzed.

- Economic Feasibility-

  Economic feasibility refers to the feasibility of the considered project to produce economic benefits. A benefit-cost analysis is needed. Furthermore, the economic feasibility of a project can also be evaluated by a breakeven analysis. In order to facilitate the consistent basis for the evaluation, the tangible and intangible facet of a project must be translated into the economic terms. Economic feasibility is critical even when the project is non-profit in nature.

- OPERATIONAL FEASIBILITY-

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirement analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

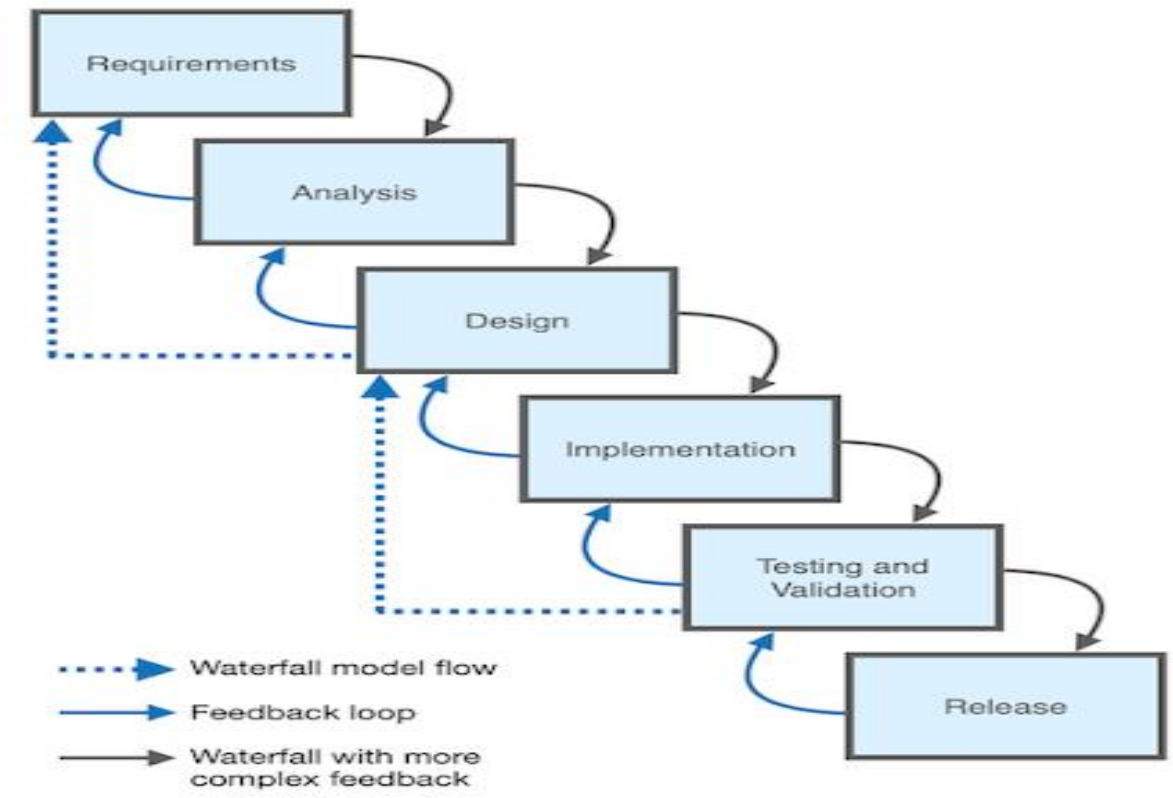
Social Feasibility-

- The affect that a proposed project may have on the social system in the project environment is addressed in the social feasibility. It may happen that particular category of employees may be short or not available as a result of ambient social structure. The influence on the social status of the participants by the project should be evaluated in order to guarantee compatibility. It must be identified that employees in the particular industries may have specific status symbols within the society.

SYSTEM PLANNING

✓ Project life cycle-

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process. WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.



The developer is responsible for:

• Developing the system, which meets the SRS and solving all the requirements of the system?

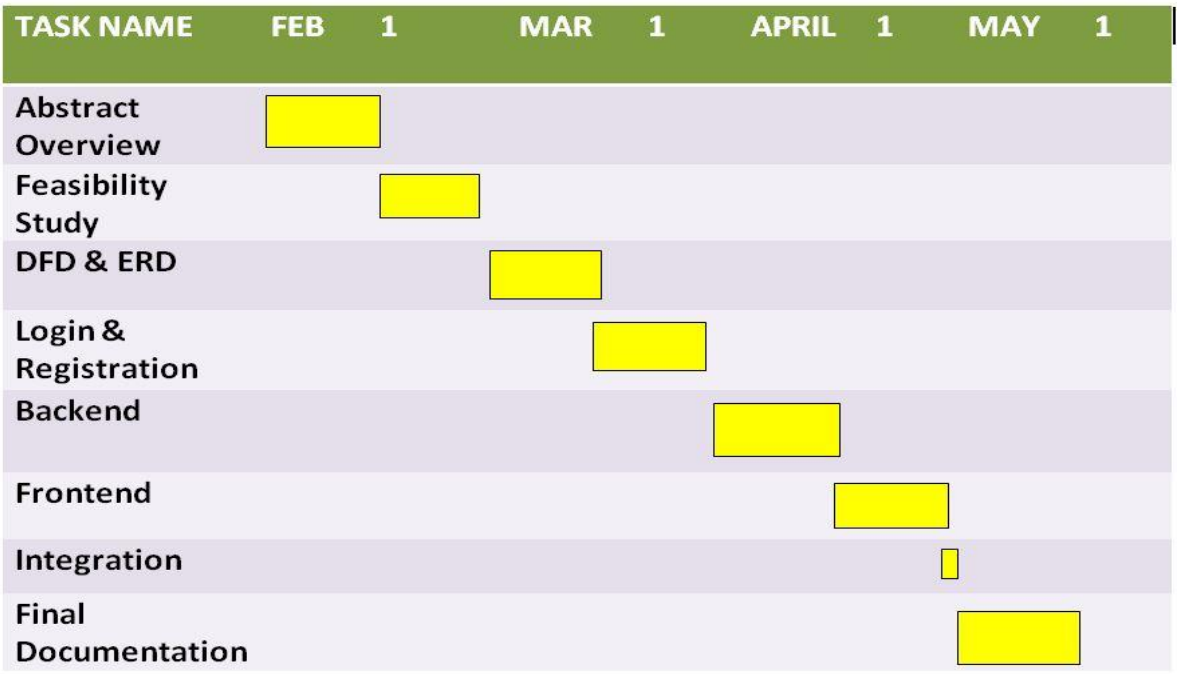
• Demonstrating the system and installing the system at client's location after the acceptance testing is successful.

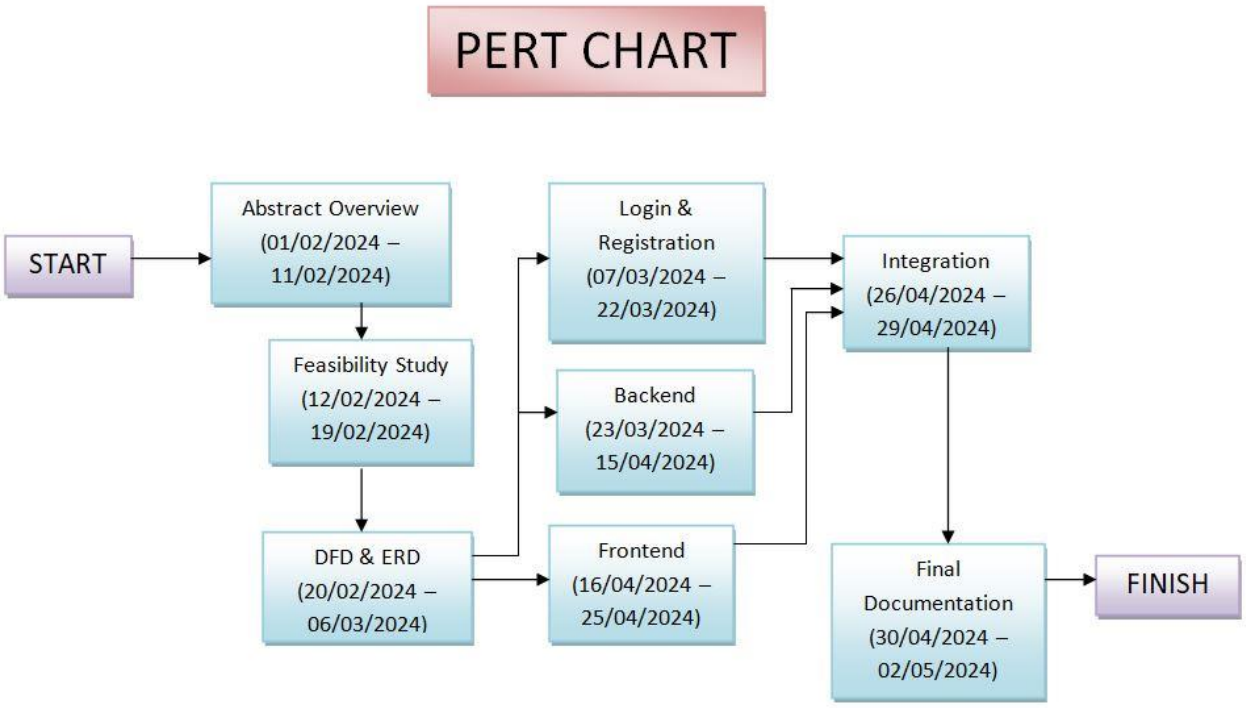
• Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.

• Conducting any user training that might be needed for using the system.

• Maintaining the system for a period of one year after installation.

GNATT CHART





SOFTWARE REQUIREMENT SPECIFICATION

❖ Functional Requirements–

- **Adding Products:** It adds/edits the product and product details to the site.
- **Checking product availability**: It always checks product's availability.
- **Add category**: It adds/edits the categories to the site.
- **Placing Order**: It is used to place an order as per the user's requirements.
- **Making Payment**: It is used to make payments for the order placed.
- **Search for product**: It is used to search for the required products.

❖ Non-functional requirements-

- Usability Requirement: The system shall allow the users to access the system from any browsers, no special training is required. The system user friendly and the system are written in simple English.
- Availability Requirement: The system is available 100% for the user and is used by24 hours a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

- Accuracy: The system should accurately provide real time information taking into consideration various issues. The system shall provide 100% access reliability.
- Performance Requirement: The information is refreshed at regular intervals depending upon whether some updates have occurred or not. The system shall respond the member in less than 2 seconds.
- Security Requirement: System will use a secured database and the system will have different users and each user has different types of constraints. Only admins have the rights to update database information of other users.
- Reliability Requirement: The system has to be 100% reliable due to the importance of data and the damages that can be caused by incorrect data. The system will run 7 days a week and 24 hours a day

Development Platform-

❖ HARDWARE REQUIREMENTS

• Computer that has a 1.6GHz or faster processor

• Minimum 512MB RAM is needed

• HDD 256GB Hard Disk Space and above hardware requirements 5400 RPM hard disk drive

❖ SOFTWARE REQUIREMENTS

• WINDOWS 10.

• MERN (MongoDB, Express, React, Node)

• Visual Studio Code

COST AND BENEFIT ANALYSIS

When a project is developed costs are incurred through its lifecycle. Benefits are realized in the form of reduced operation costs, improved corporate image, staff efficiency or revenues.

Cost and benefit analysis is a procedure that gives a picture of the various costs, benefits & rules associated with a system. The determination of costs & benefits entails the following steps.

1. Costs & benefits pertaining to the project are identified.
2. The various costs & benefits are categorized for analysis.
3. An evaluation method is selected.
4. The results of analysis are interpreted.

Costs are identified as tangible, intangible costs, direct and indirect costs, fixed and variable costs.

Cost Estimation of the Project:

- We are using COCOMO in this project to calculate the cost estimation.

- COCOMO is one of the most widely used software estimation models in the world
- It was developed by Barry Boehm in 1981
- COCOMO predicts the effort and schedule for a software product development based on inputs relating to the size of the software and a number of cost drivers that affect productivity.

- COCOMO has three different models that reflect the complexity:
  - The Basic Model
  - The Intermediate Model
  - And the Detailed Model

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

- We are using Basic COCOMO as Organic for the rough estimation of the Project Cost.
- A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved

in the past and also the team members have a nominal experience regarding the problem.

- Basic COCOMO model formula:
- Effort = a*(KLOC)^b person-month
- TDev = c*(Effort)^d month
- Person required = Effort/TDev
- Here TDev is the time required to develop the software.
- The total estimated lines of code for the project = 51,963 LOC. = 51.963 KLOC
- Effort = 2.4*(51.963)^1.05 = 151.95 person-month
- TDev = 2.5*(151.95)^0.38 = 16.86 month
- Person required = 151.95/16.86 = 10 (approx.)

SOFTWARE DESIGN SPECIFICATION

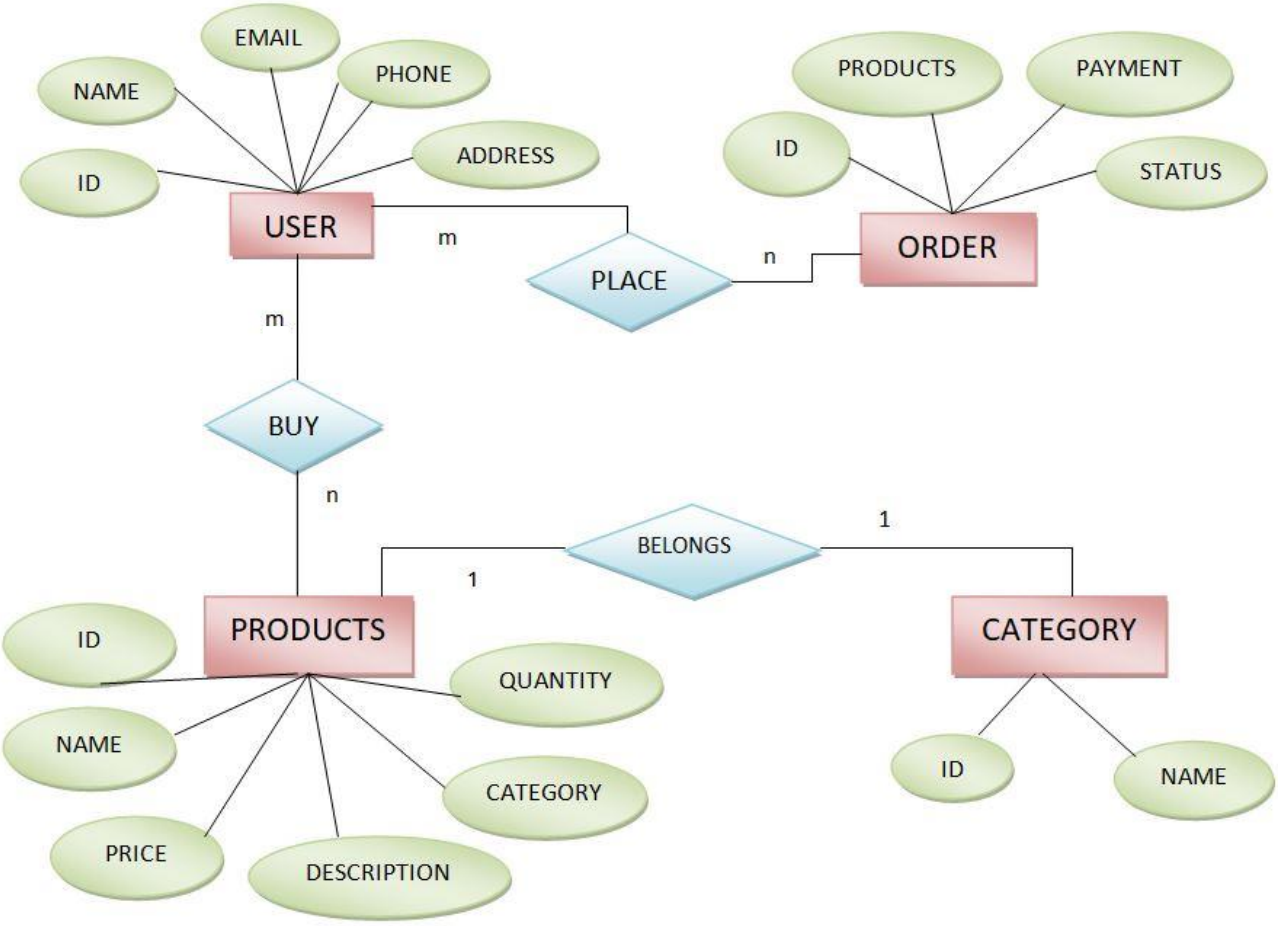
- Entity Relationship Diagram-

  In software engineering, an entity–relationship model (ER model) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them. However, variants of the idea existed previously, and have

been devised subsequently such as super type and subtype data entities and commonality relationships.

- An entity–relationship model is a systematic way of describing and defining a business process. The process is modelled as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entities may have various properties (attributes) that characterize them. Diagrams created to represent these entities, attributes, and relationships graphically are called entity– relationship.

E-R DIAGRAM



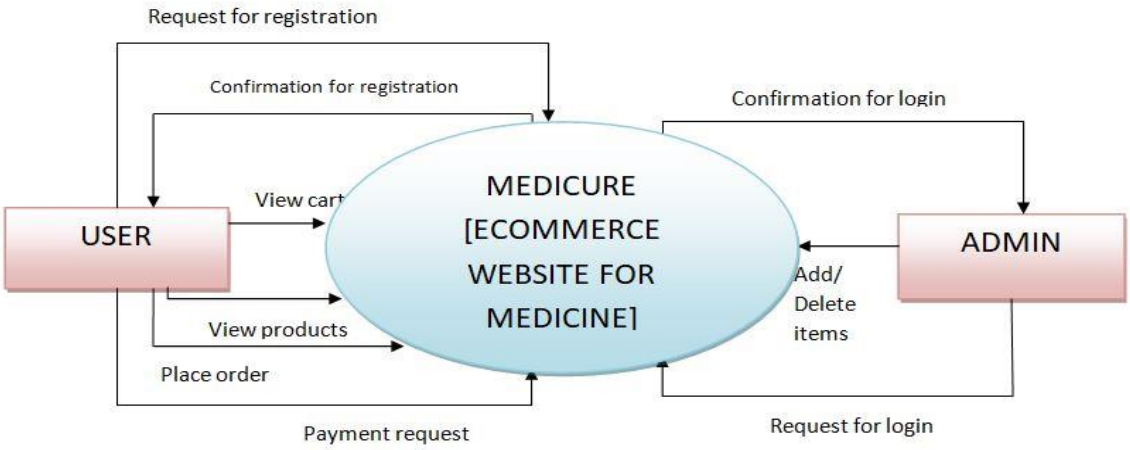
Data Flow Diagram-

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. A structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process-oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

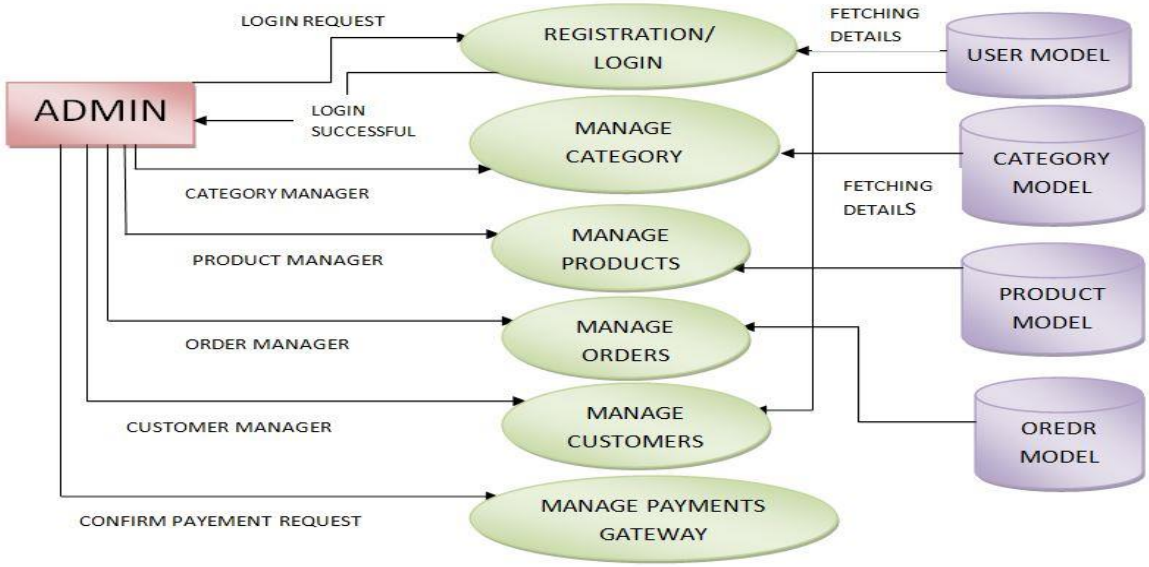
0 Level DFD-The 0 level dfd known as **context level** data flow diagram. The context level data flow diagram (dfd) describes the whole system. The (o) level dfd describes all user modules who run the system. Below context level data flow diagram of Online Medical store [Medicure] project shows the one Admin user can operate the system. Admin do all activities after login to system.

0 LEVEL DFD

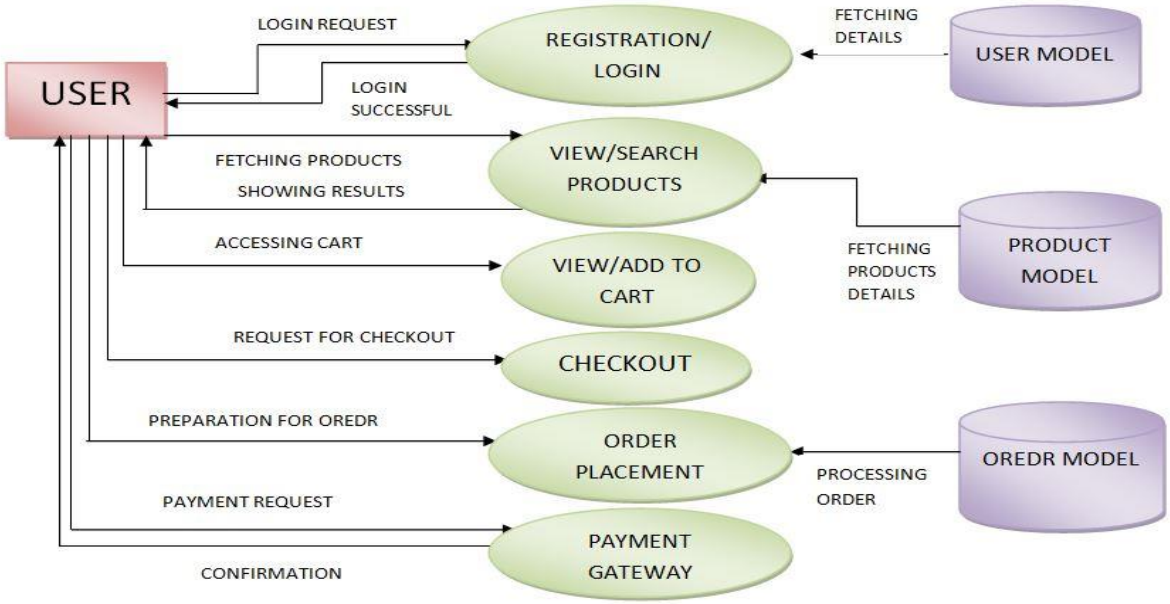


1st level DFD- The Admin side DFD describe the functionality of Admin. Admin is a responsible person who run the project. After login to system admin can first add product and admin can manage orders and order- payment details.

LEVEL 1 DFD FOR ADMIN



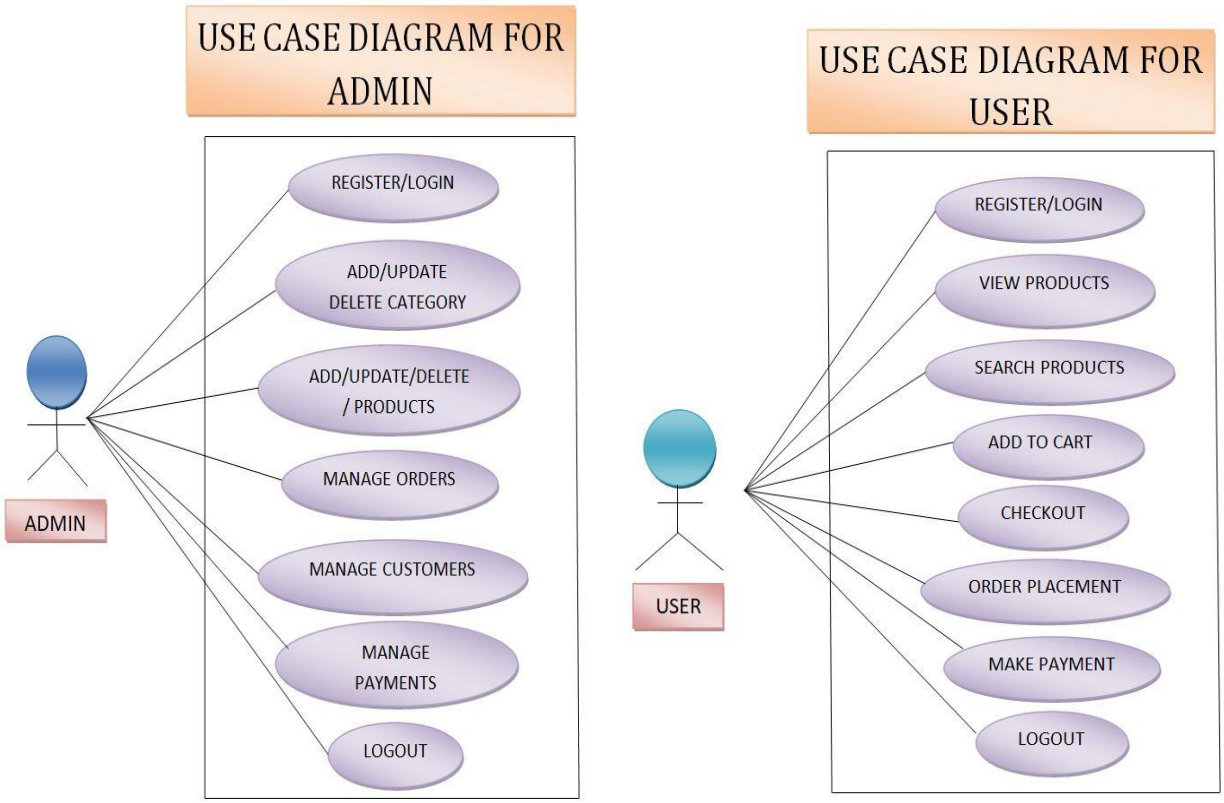
LEVEL 1 DFD FOR USER



Use Case Diagram -

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.



DataBase Schema:

USER TABLE					
NAME	TYPE	NULL	DEFAULT	UNIQUE	REQUIRED
ID	STRING	NO	N/A	YES	YES
NAME	STRING	NO	N/A	NO	YES
EMAIL	STRING	NO	N/A	YES	YES
PASSWORD	STRING	NO	N/A	NO	YES
PHONE	STRING	NO	N/A	YES	YES
ADDRESS	STRING	NO	N/A	NO	YES

PRODUCT TABLE					
NAME	TYPE	NULL	DEFAULT	UNIQUE	REQUIRED
ID	STRING	NO	N/A	YES	YES
NAME	STRING	NO	N/A	YES	YES
DESCRIPTION	STRING	NO	N/A	NO	YES
PRICE	NUMBER	NO	N/A	NO	YES
CATEGORY	STRING	NO	N/A	NO	YES
QUANTITY	NUMBER	NO	0	NO	YES

ORDER TABLE					
NAME	TYPE	NULL	DEFAULT	UNIQUE	REQUIRED
ID	STRING	NO	NONE	YES	YES
PRODUCTS	STRING	NO	NONE	NO	YES
PAYMENT	STRING	NO	NONE	YES	YES
BUYER	STRING	NO	NONE	NO	YES
STATUS	STRING	NO	ORDER PLACED	NO	YES

CATEGORY TABLE					
NAME	TYPE	NULL	DEFAULT	UNIQUE	REQUIRED
ID	STRING	NO	NONE	YES	YES
NAME	STRING	NO	NONE	YES	YES

Methodology used for testing and some testing results:

Testing strategies and test plan-

The objective of the testing is to test the written program. Testing is done at various levels to ensure the quality of the software developed. The user tests the system to verify that the system function as specified. Once it is verified the system is ready for lunch in web. So, we test our product through different methods of testing. These are: -

Black Box Testing:

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

Specification-based testing:-

> Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.
>
> Specification-based testing is necessary, but it is insufficient to guard against certain risks.

White Box Testing:

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

Types of white box testing:

The following types of white box testing exist:

API testing (application programming interface) - Testing of the application using Public and Private APIs

Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)

Fault injection methods - improving the coverage of a test by introducing faults to test code paths.

Test coverage:

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

Function coverage, which reports on functions executed

Statement coverage, which reports on the number of lines executed to complete the test

They both return a code coverage metric, measured as a percentage.

TEST CASE AND TEST RESULTS:

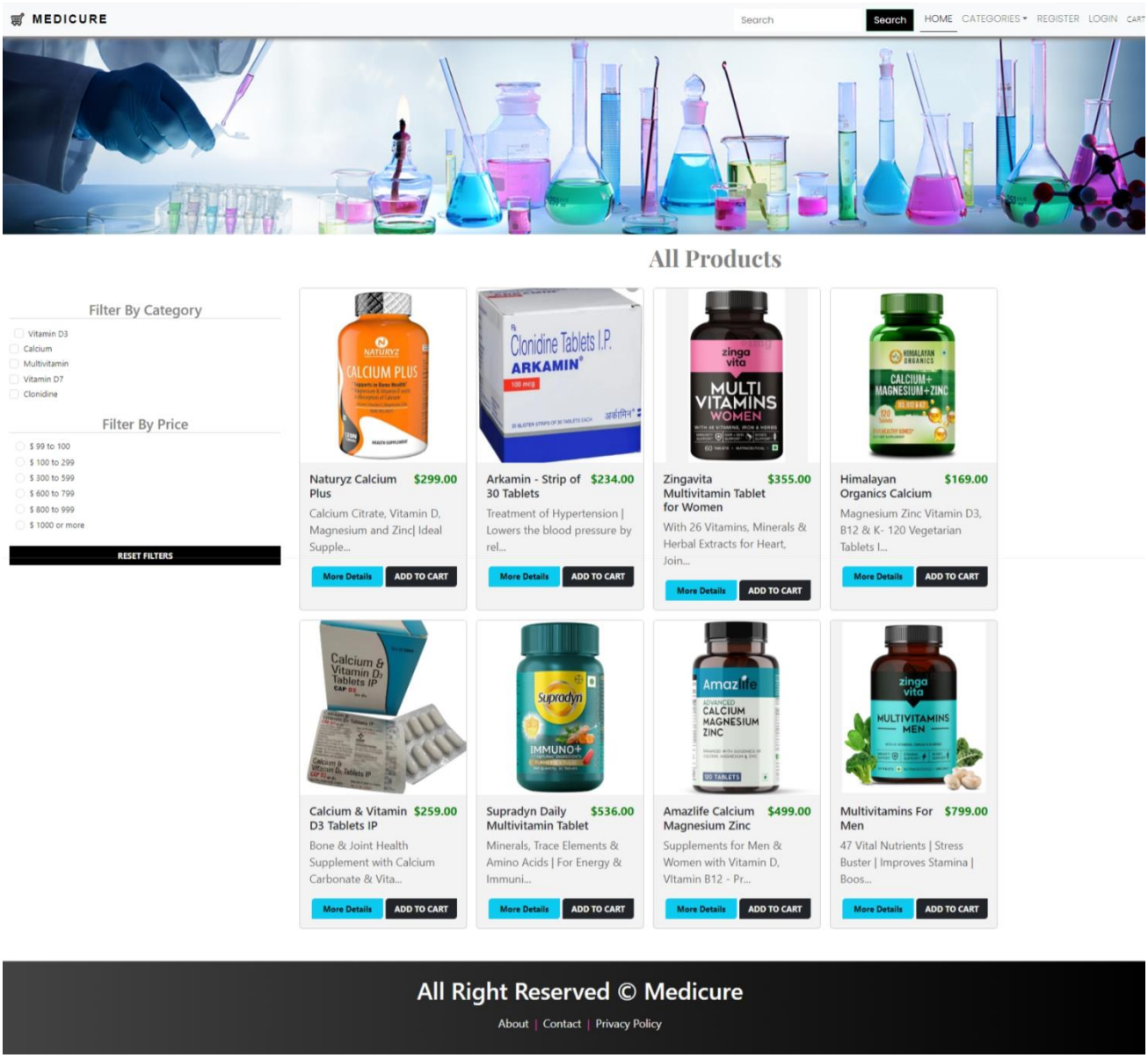
Test cases	Expected outcome	Exact outcome	Result
1. Login using correct credentials.	Login successful.	Login successful.	Pass
2. Attempt to login using wrong password	Login failed with error message “Invalid password”	Login failed with error message “Invalid password”	Pass
3. Attempt to login using wrong email	Login failed with error message “Please register”	Login failed with error message “Please register”	Pass
4. Attempt to login without filled up email field	“Email is required” message will be displayed.	“Email is required” message is displayed.	Pass
5. Attempt to login without filled up the password field	“Password is required” message will be displayed.	“Password is required” message is displayed.	Pass

6. Attempt to register using registered email.	Login failed with error message “Already registered please login”	Login failed with error message “Already registered please login”	Pass
7. Attempt to register without filled up name field	“Name is required” message will be displayed.	“Name is required” message is displayed.	Pass
8. Attempt to register without filled up email field	“Email is required” message will be displayed.	“Email is required” message is displayed.	Pass
9. Attempt to register without filled up Set Password field	“Password is required” message will be displayed.	“Password is required” message is displayed.	Pass
10. Register using unregisters email.	Registration done with message “Registration successful.”	Registration done with message “Registration successful.”	Pass
11. Login as a customer	List of available medicine gets displayed.	List of available medicine gets displayed.	Pass
12. Search for a medicine which is available	Searched item gets displayed	Searched item gets displayed	Pass
13. Search for a medicine which is not available	Searched item is not found	Searched item is not found	Pass
14.Filter items by specific category	Item of specified category gets displayed	Item of specified category gets displayed	Pass
15.Filter items by specific range of price	Item with specified price range gets displayed	Item with specified price range gets displayed	Pass
16. Add Item to Cart	Item is added successfully.	Item is added successfully.	Pass
17. Remove from Cart	Item is removed successfully	Item is removed successfully	Pass
18. Checkout without login credential	Payment option gets displayed	Payment option gets displayed	Pass
19. Checkout with login	“Please login to checkout” button gets display and	“Please login to checkout” button gets display and	Pass

credential	payment button is disabled.	payment button is disabled.	
20. Check orders List	List of ordered items gets displayed.	List of ordered items gets displayed.	Pass
21. Cancel Order	Order is cancelled successfully and this status will be modified in Admin dashboard.	Order is cancelled successfully and this status is modified in Admin dashboard.	Pass
22. Attempt to add create category	Category is added successfully.	Category is added successfully.	Pass
23. Attempt to edit category	Category is updated successfully.	Category is updated successfully.	Pass
24. Attempt to delete category	Category is deleted successfully.	Category is deleted successfully.	Pass
25. Attempt to add product information as an admin	Product is added successfully.	Product is added successfully.	Pass
26. Attempt to delete product	Product is deleted successfully.	Category is deleted successfully.	Pass
27. Check the list of added products	List of added products gets displayed.	List of added products gets displayed.	Pass
28. Attempt to delete product	Product is deleted successfully.	Product is deleted successfully.	Pass
29. Attempt to Change order status as admin	Order status will be changed and mail will be triggered to the customer.	Product status is changed and mail is triggered to the customer.	Pass
30. Attempt to cancel order as customer	Order status will be changed as cancel and this will also ref.	Product status gets changed and mail is triggered to the customer.	Pass
31. Check the list of customers	List of customer gets displayed.	List of customer gets displayed.	Pass
32. Click on logout	Logged out successfully.	Logged out successfully.	Pass

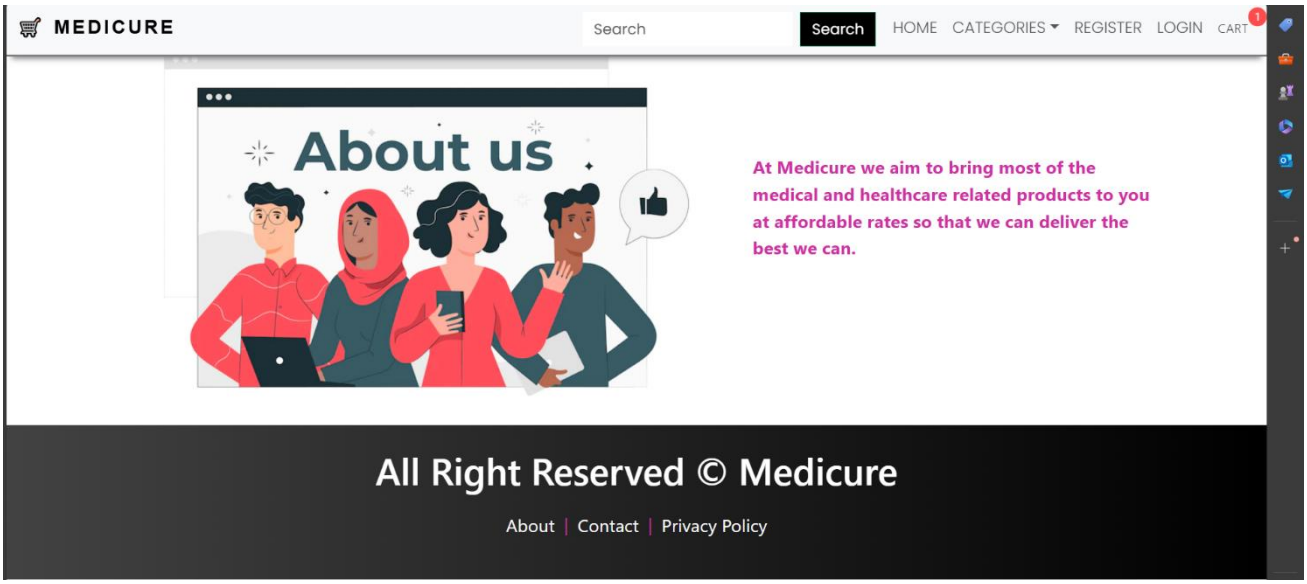
User Manual

Home page:



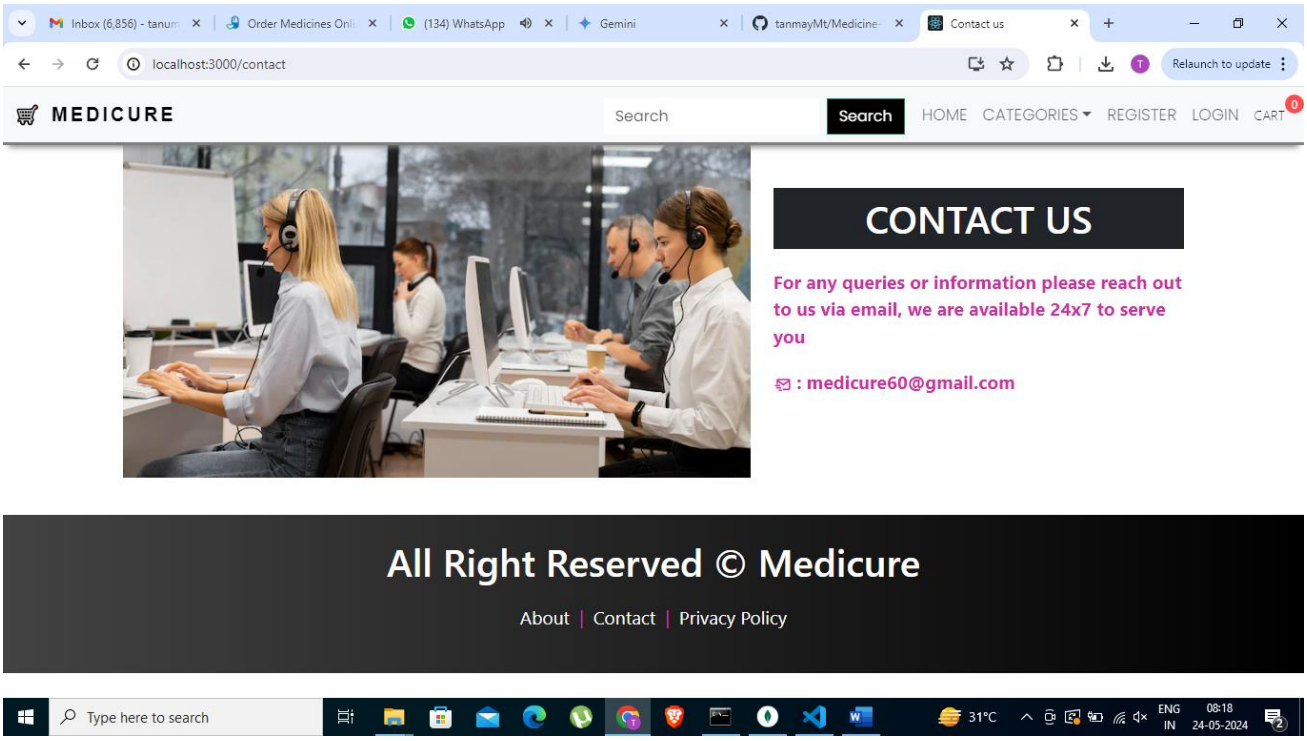
This is the landing page of the website that displays the list of available products and the option to filter the items as per the requirement and the options to login/register. Users can search the products using the Search option, user can go to the car by clicking on the Cart.

About us page:



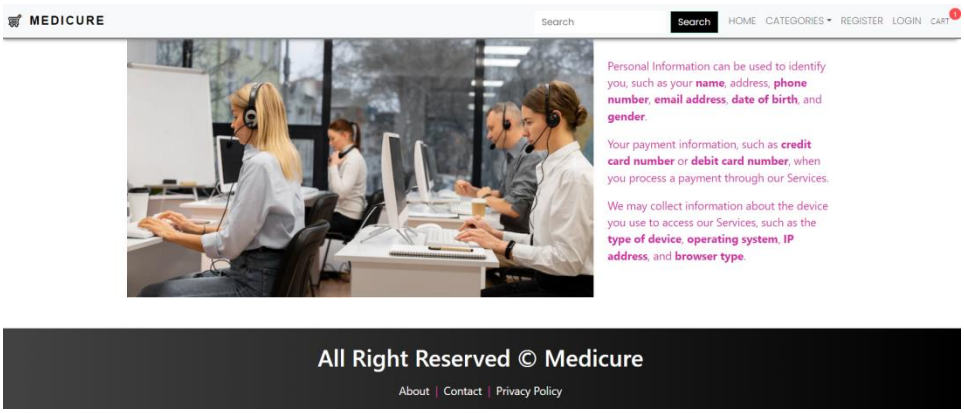
This page contains brief details about our site.

Contact page:



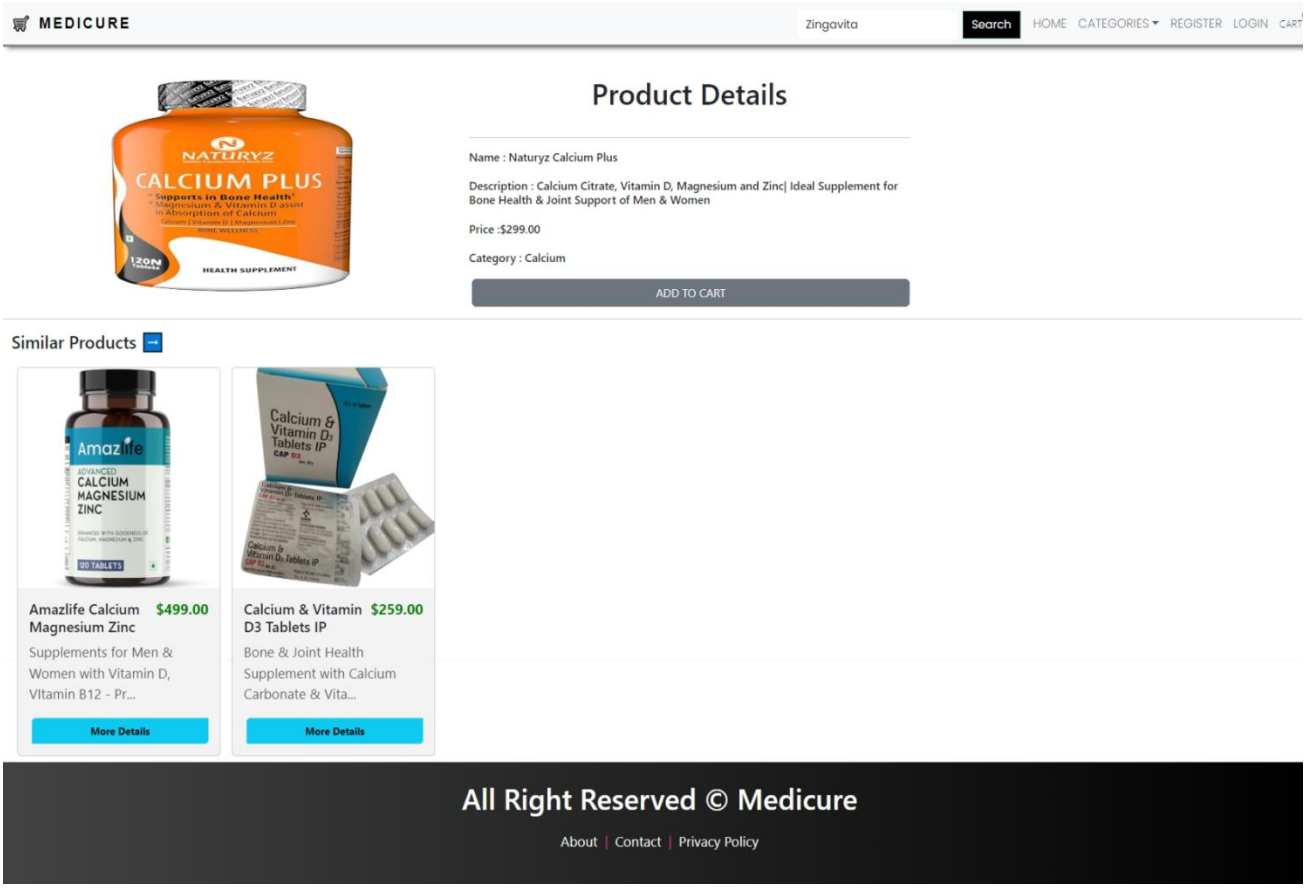
This page displays the contact information of our site.

Privacy Policy Page:



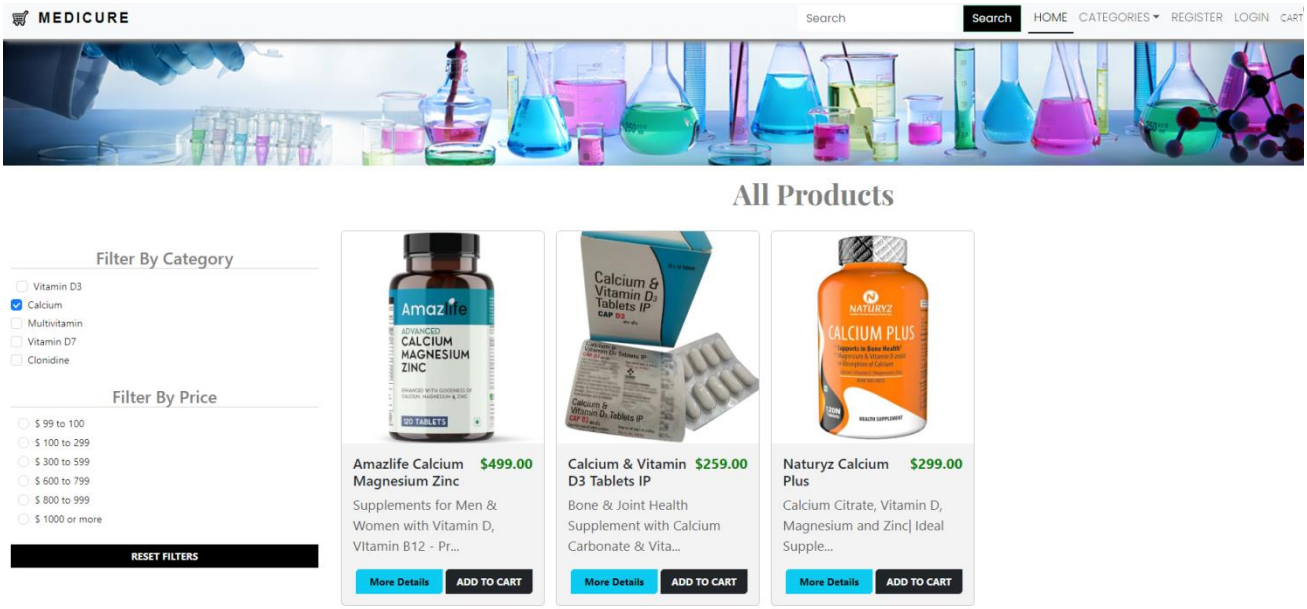
This page contains the privacy policy of the site.

Product details page:



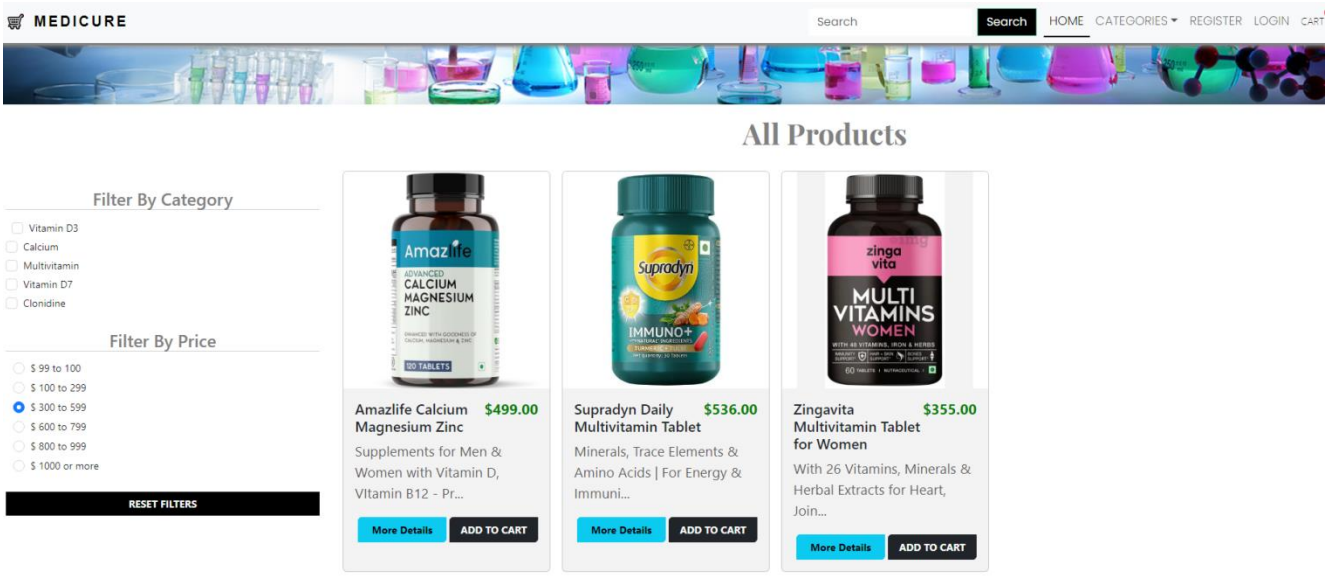
This page displays the product details along with similar items, from this page users can add the products to their cart by clicking on “Add to Cart” button.

Filter by Category:



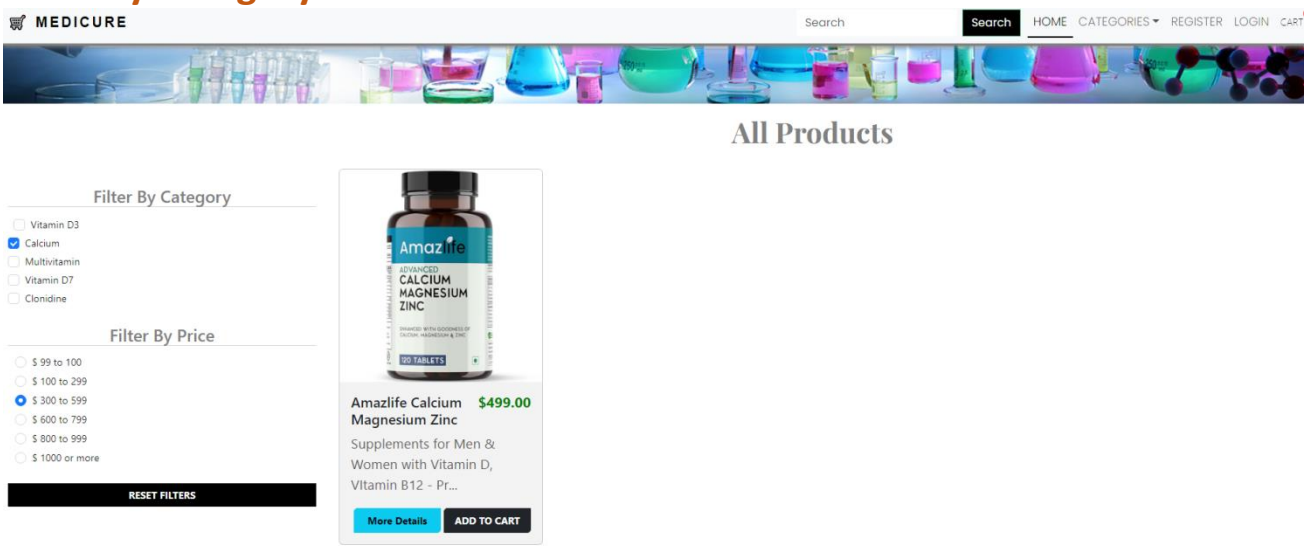
This filters the products as per the selected categories. Users can filter as per the price range and categories.

Filter by Price:



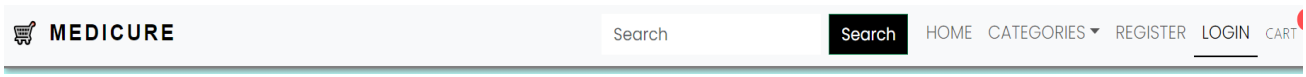
This filters the items as per the selected price range.

Filter by Category and Price:



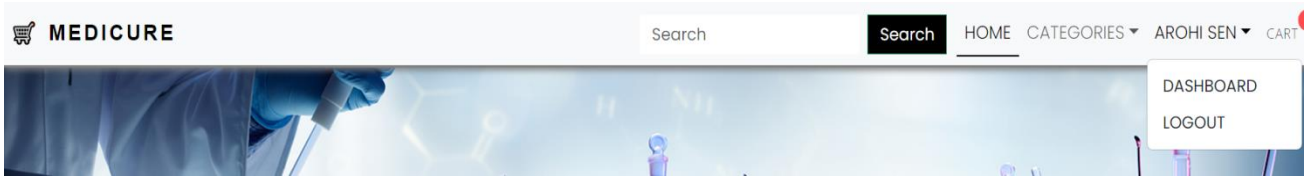
This filters the products as per the selected criteria.

Navigation Menu view while user is not logged in :



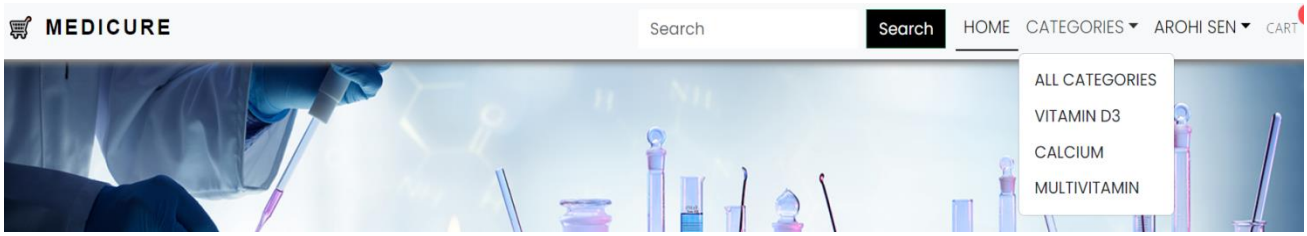
This displays the options to login/register into the system.

Navigation Menu view while user is logged in:



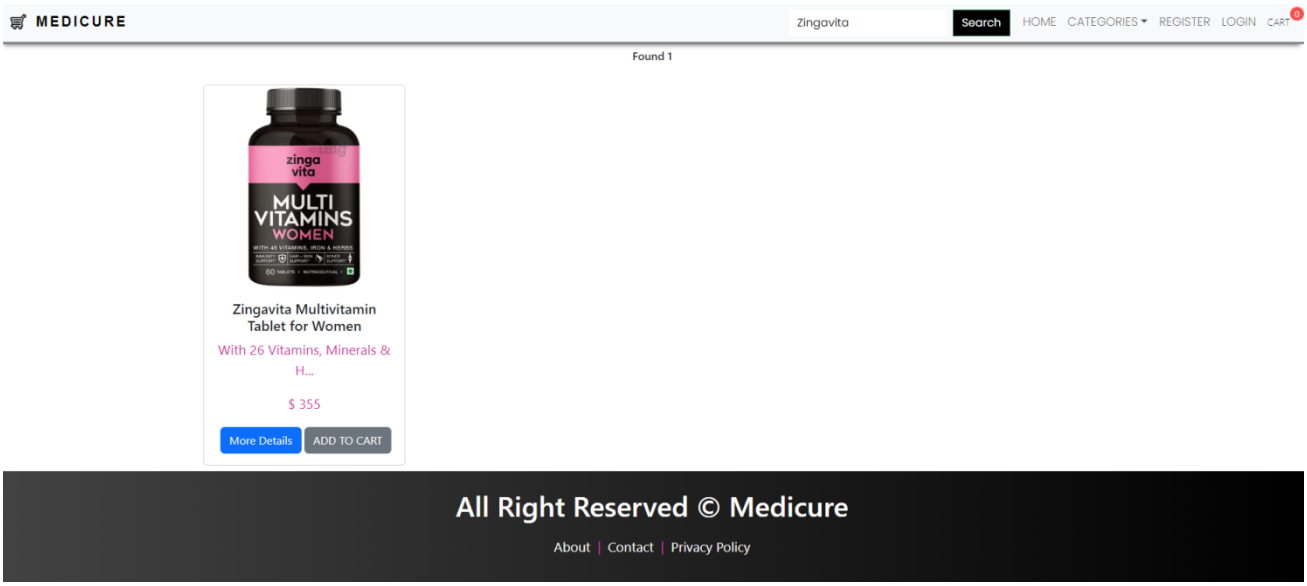
This displays the logged in user details and options to go to dashboard as well to logout from the system.

Category Dropdown:



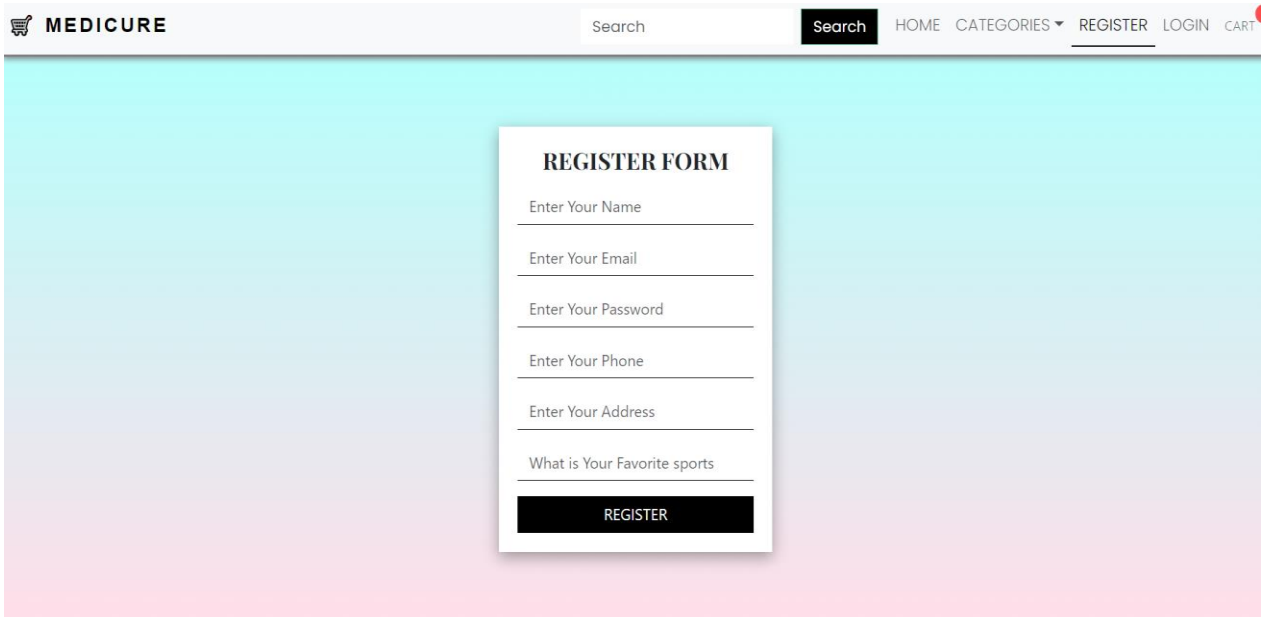
This displays the categories of the products that are available.

Search Bar:



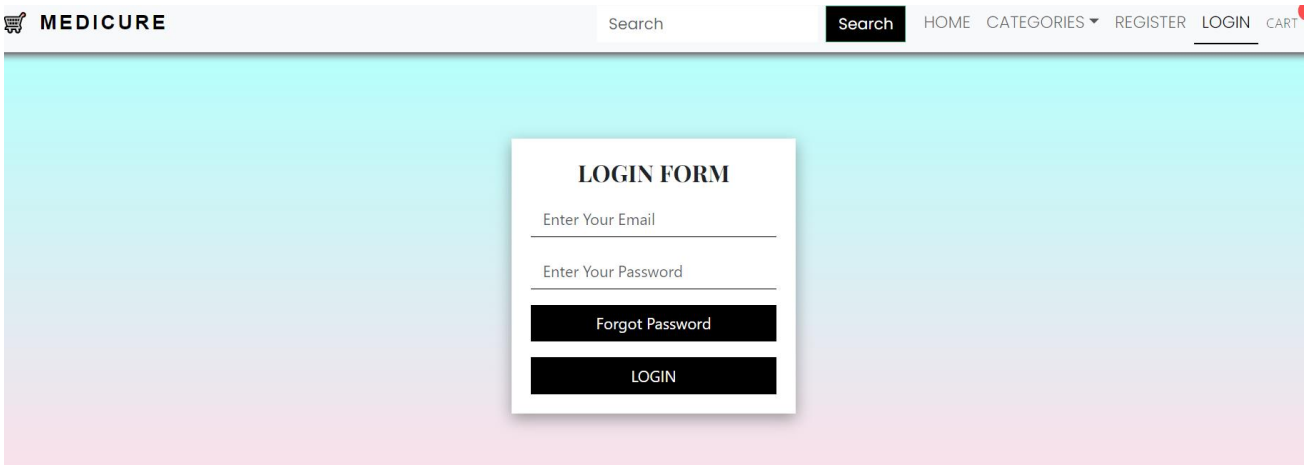
This allows the user to search for the required products.

Register Page:



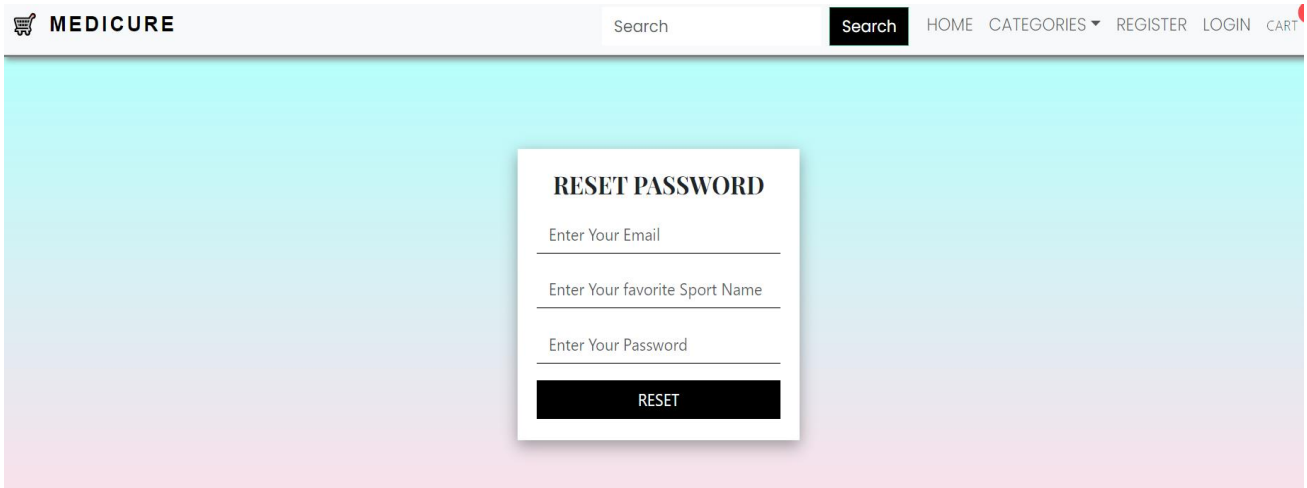
This page is used to register into the site. The address in this form will be used for delivery purpose.

Login Page:



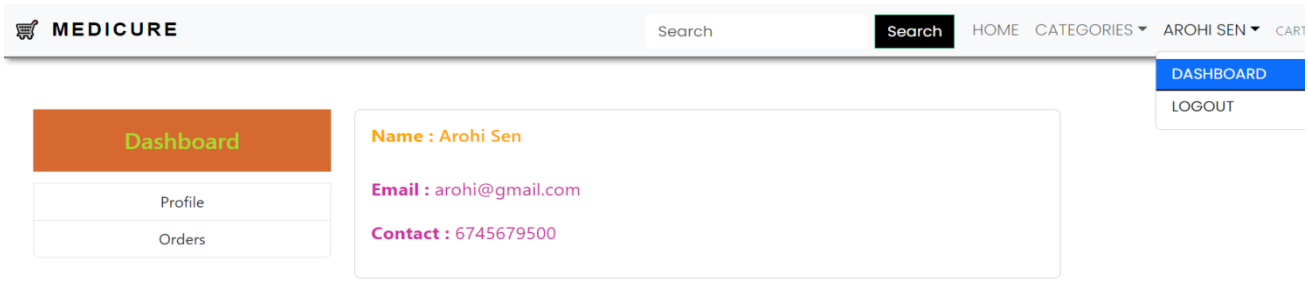
This page is used to login into the site, by clicking on “Forgot Password” customers can reset their account password.

Reset Password Page:



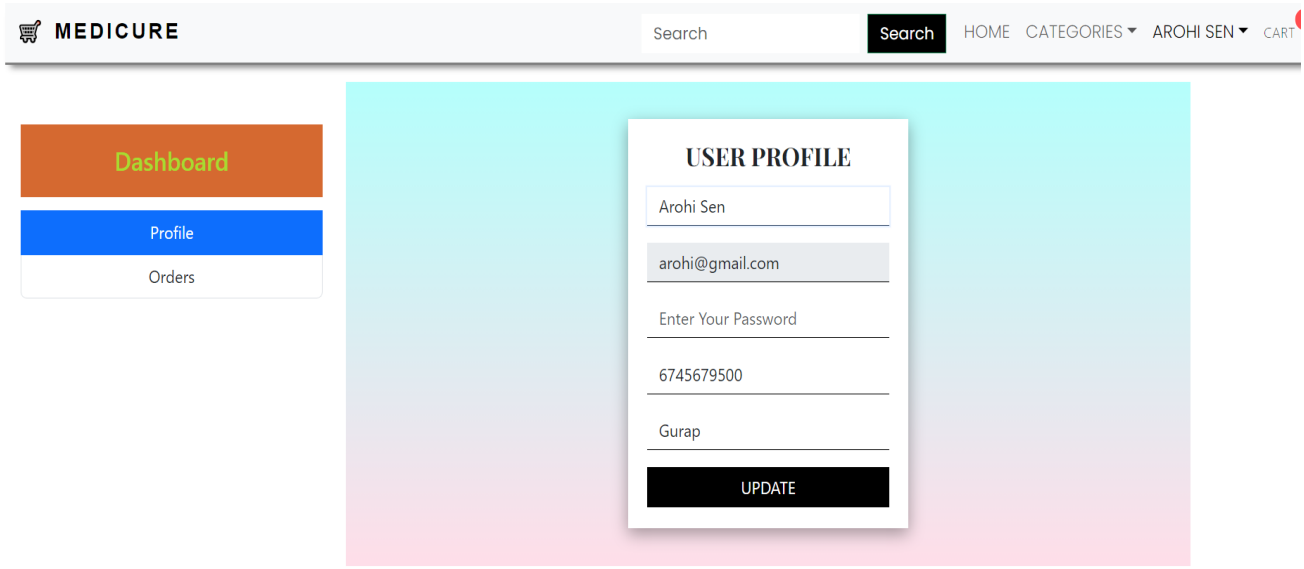
This page is used to reset the account password.

User Dashboard Page:



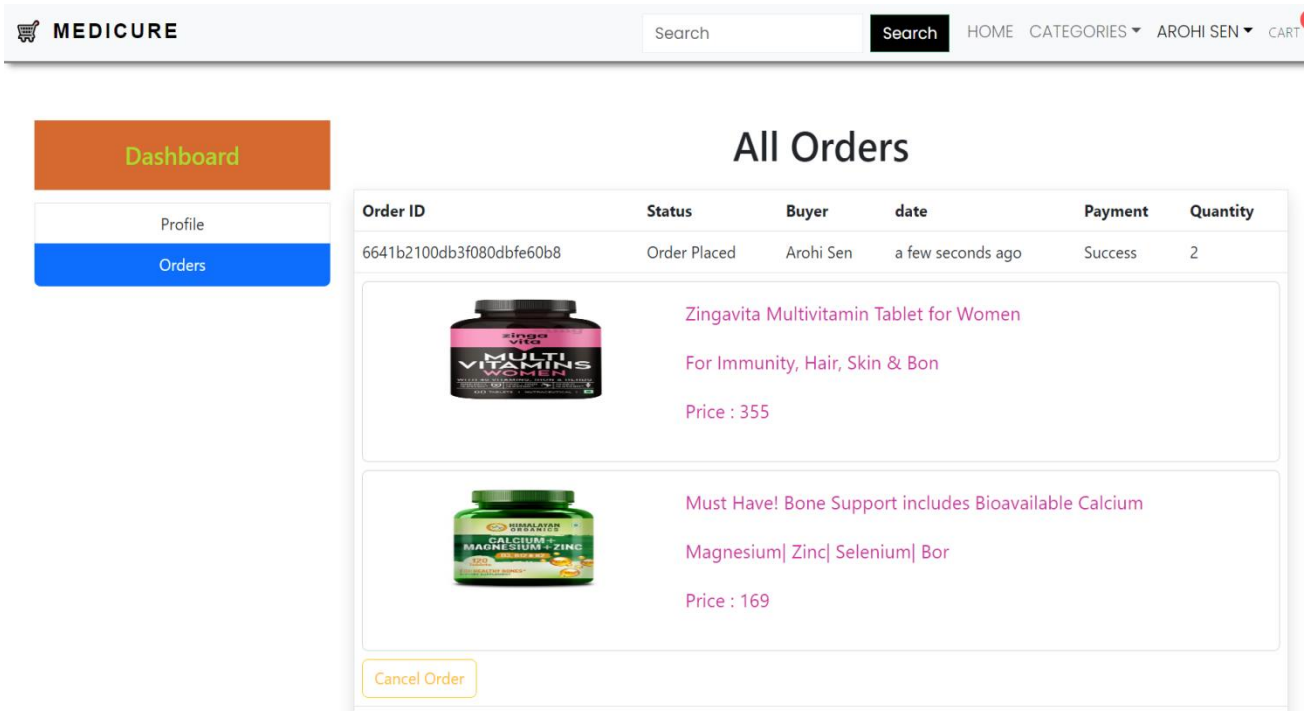
This is the user dashboard that displays the basic user details.

User Profile Page:



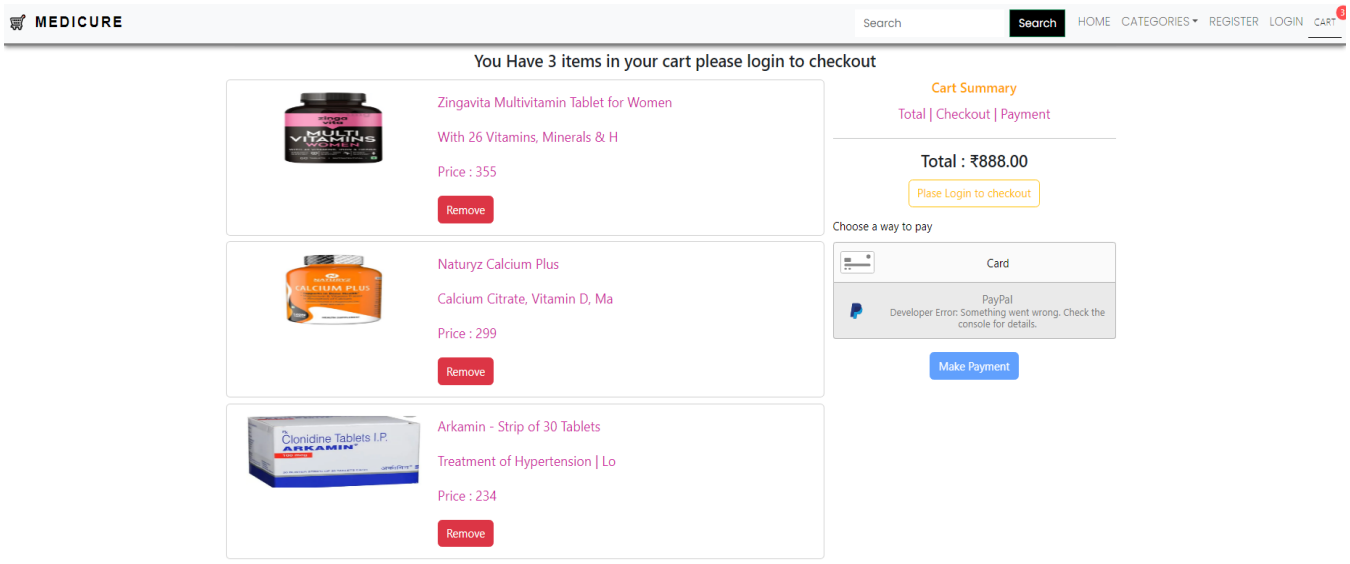
This page is used to edit the user profile and update the delivery address.

User Order page:



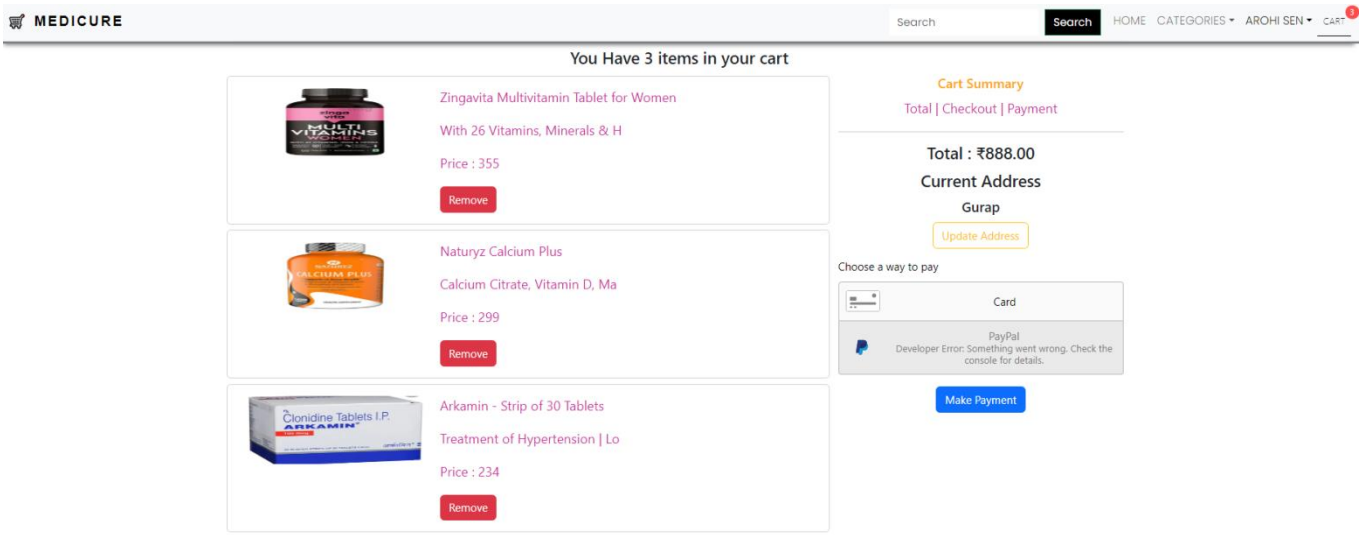
This page displays the orders that are already placed by that logged in user and from this page user can cancel an order till the order status is “Order Placed” by clicking on the “Cancel Order” button corresponding to each order.

Cart Page without Login Information:



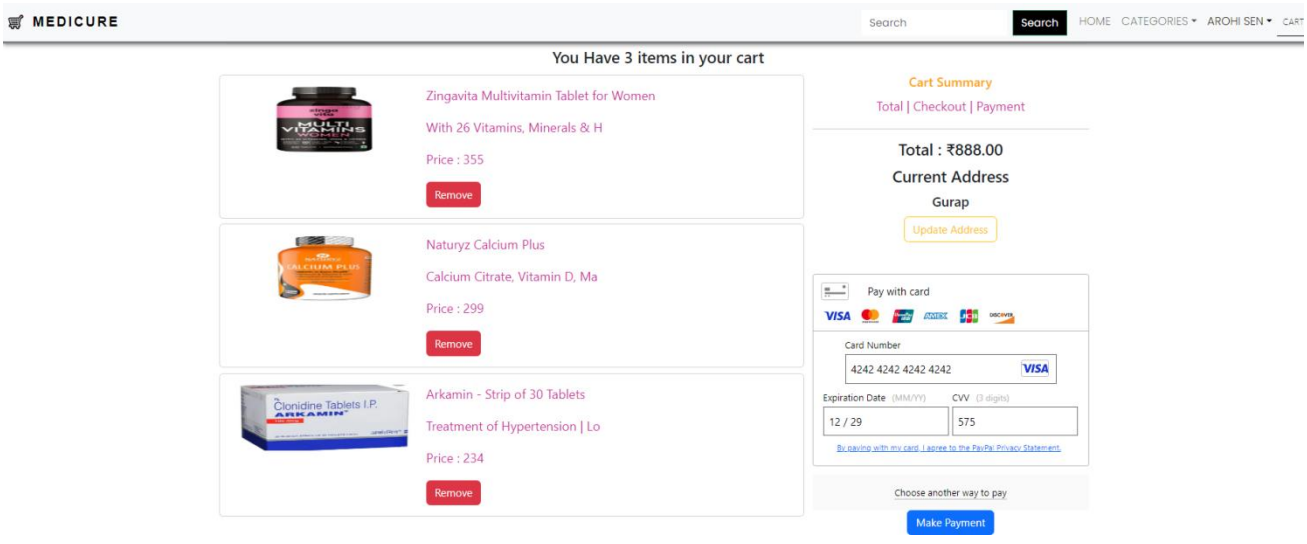
This page displays the cart information and restricts the user to checkout as the user has not logged in yet by disabling the “Make Payment” button.

Cart Page with Login Information:



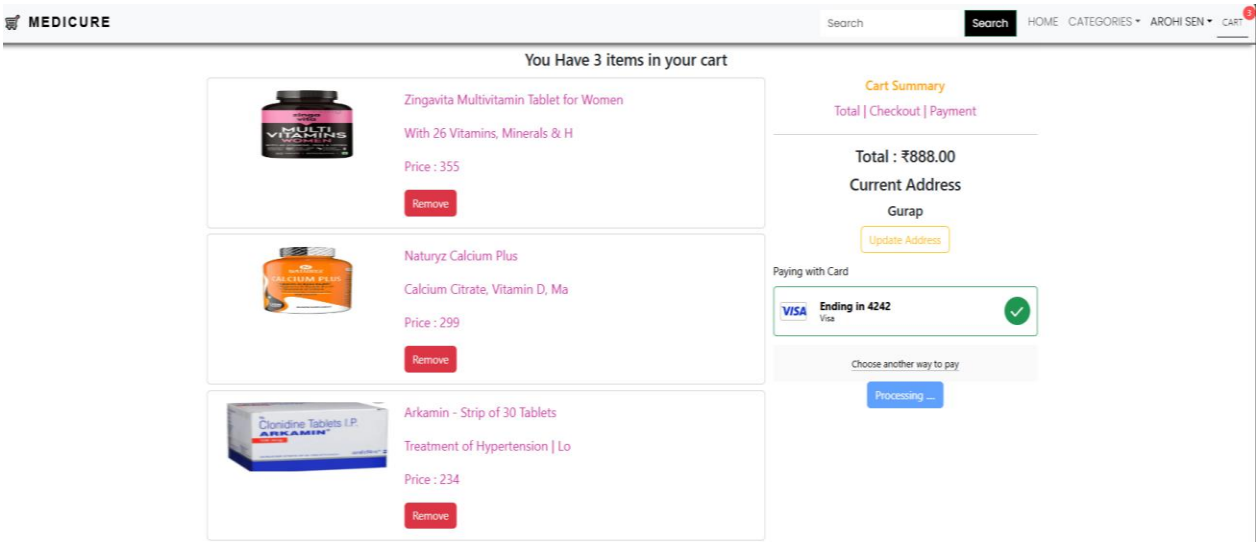
This page displays the cart that contains the products that are added by the user and allows the user to checkout after making the payment. Clicking on the “Remove” button removes the item from the cart.

Cart Page with Payment Information:



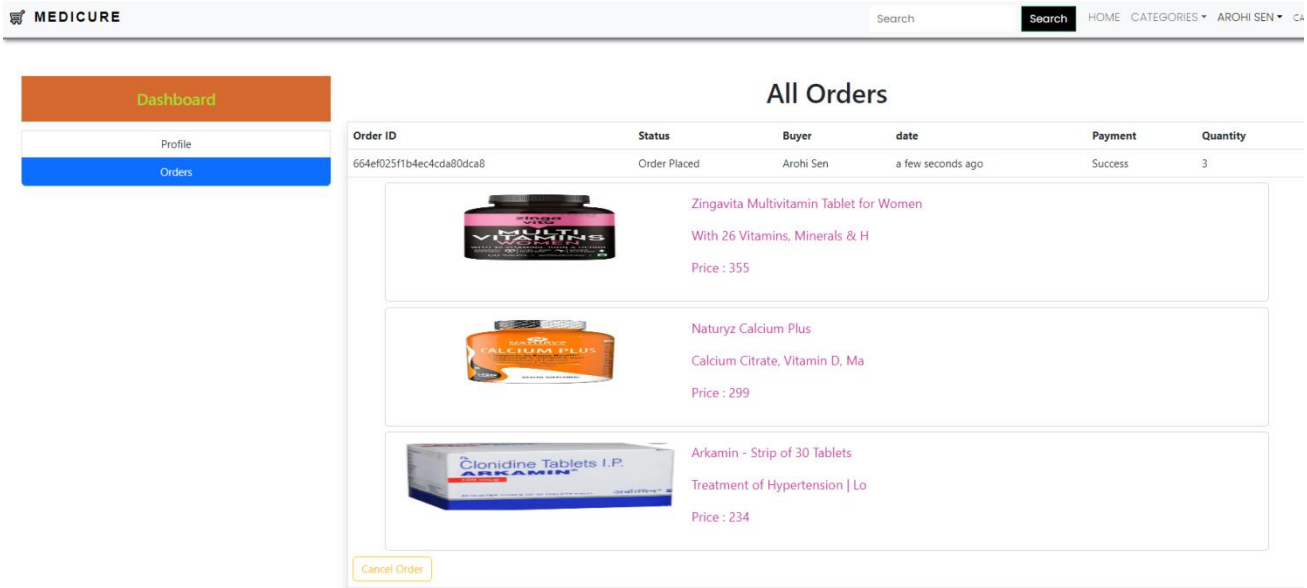
Users can make the payment for the items in their cart.

Payment in Progress:



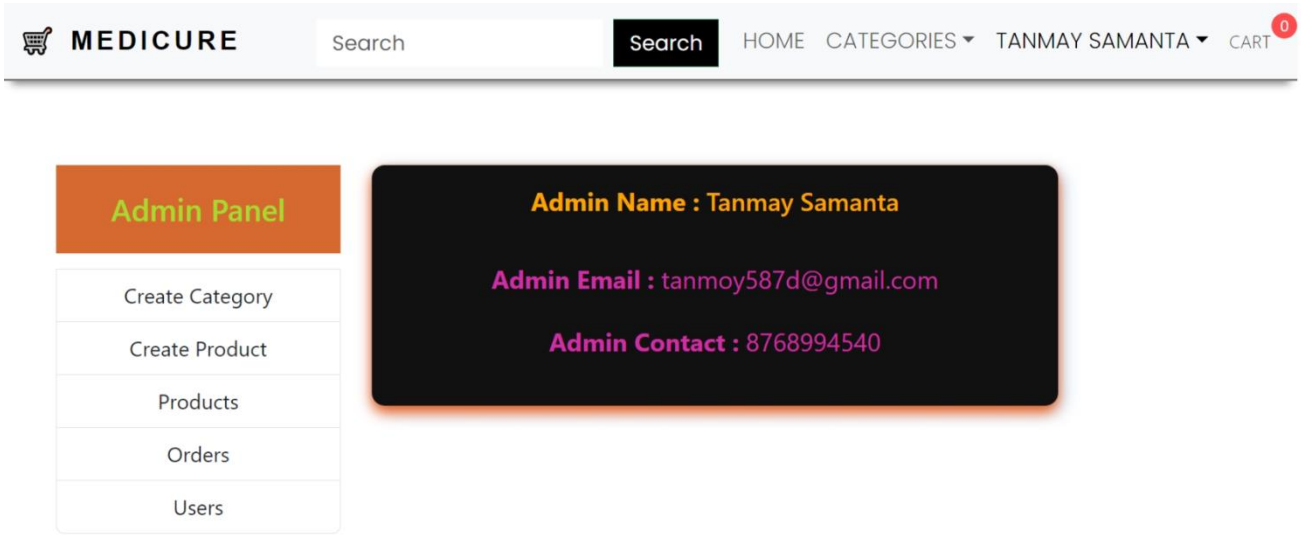
Once the payment details are verified by the system, system generates the order and send an email notification to the user.

Orders Details:



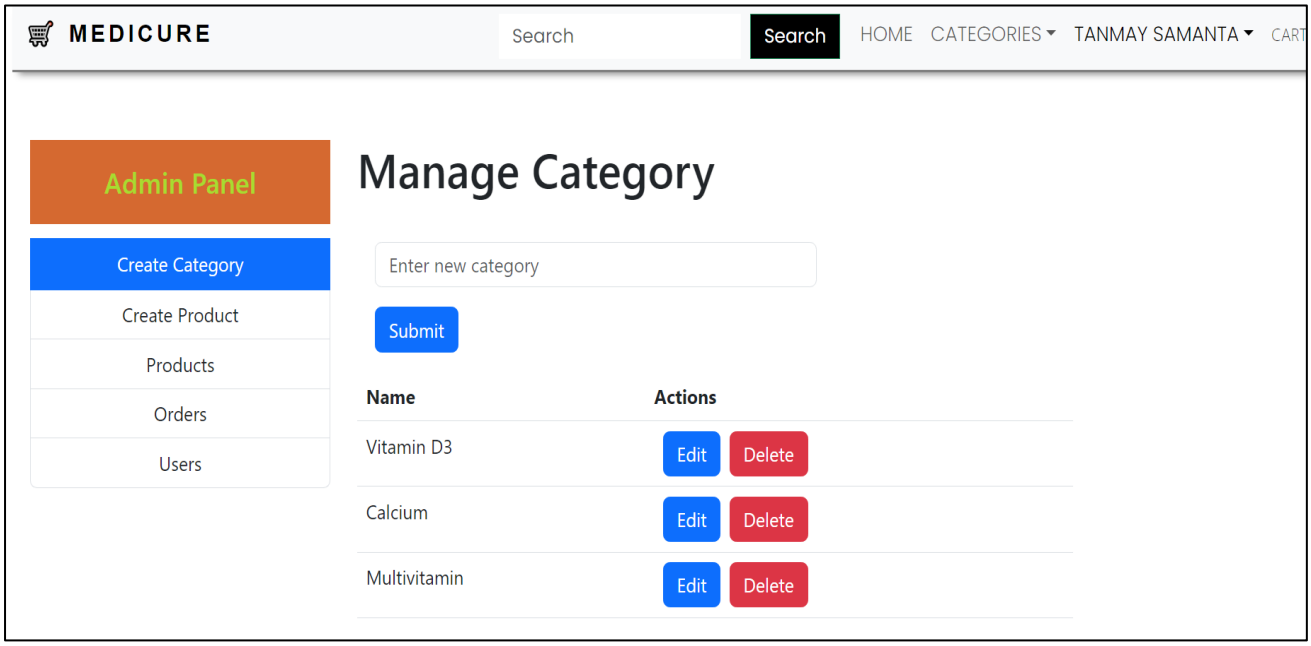
This page displays the orders that are already placed by that logged in user and from this page user can cancel an order till the order status is “Order Placed”.

Admin Dashboard Page:



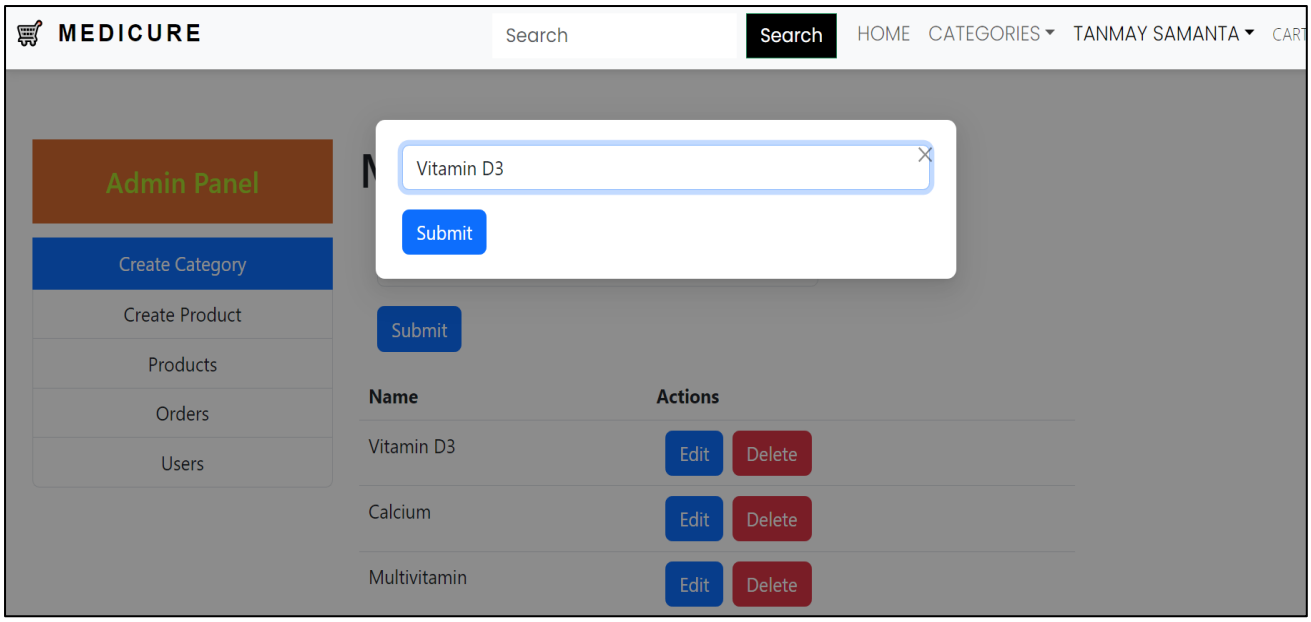
This page displays the admin information.

Admin Create Category Page:

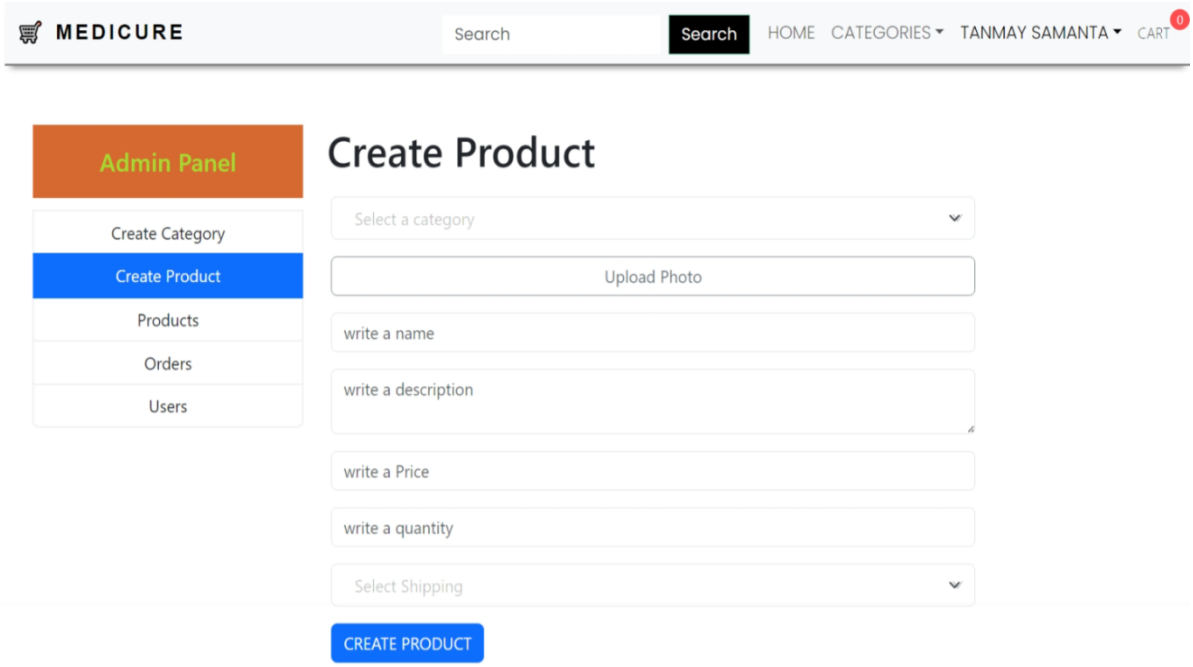


From this page, admin user can create, edit and delete categories.

Editing Category:

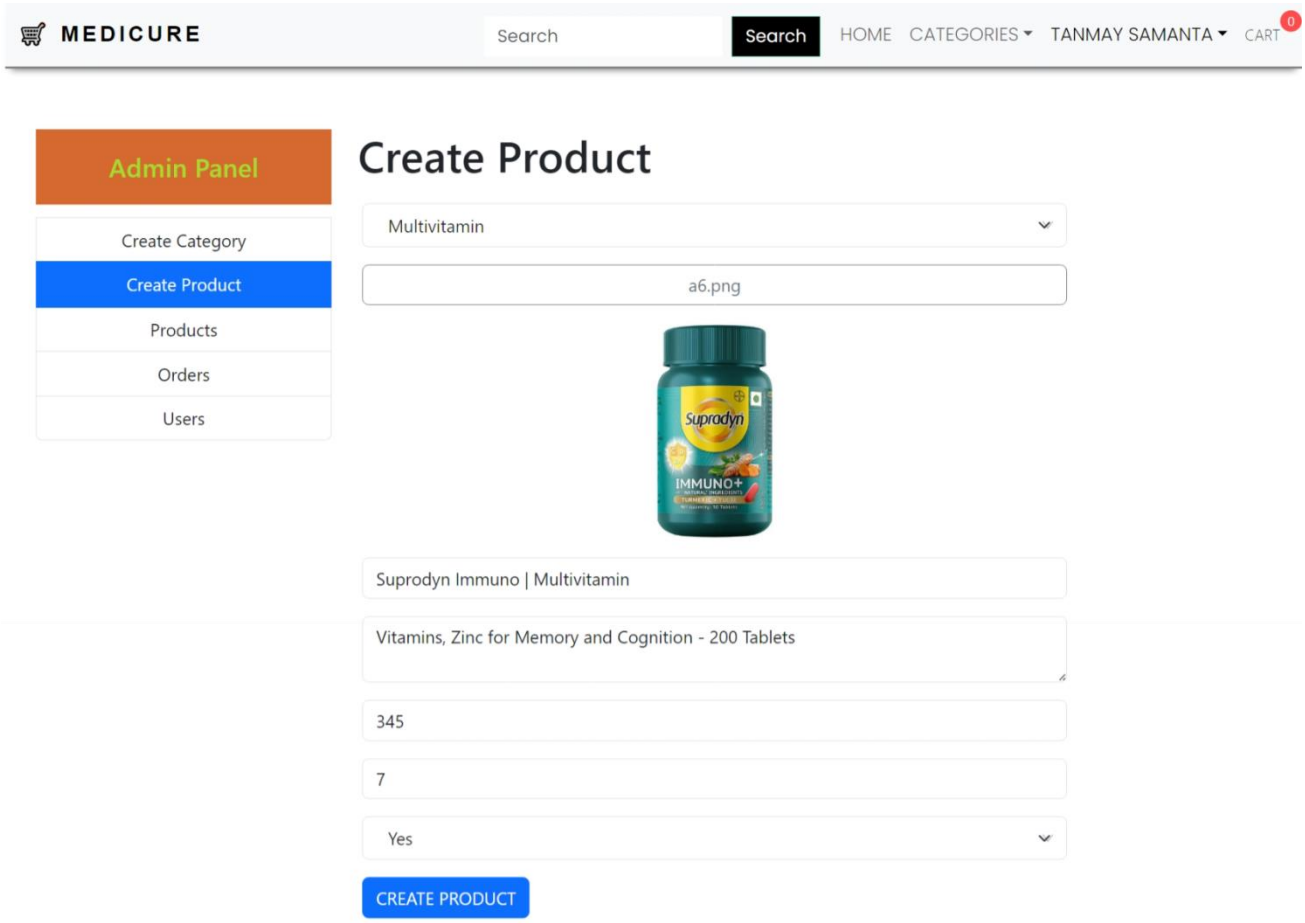


Admin Create Product Page:



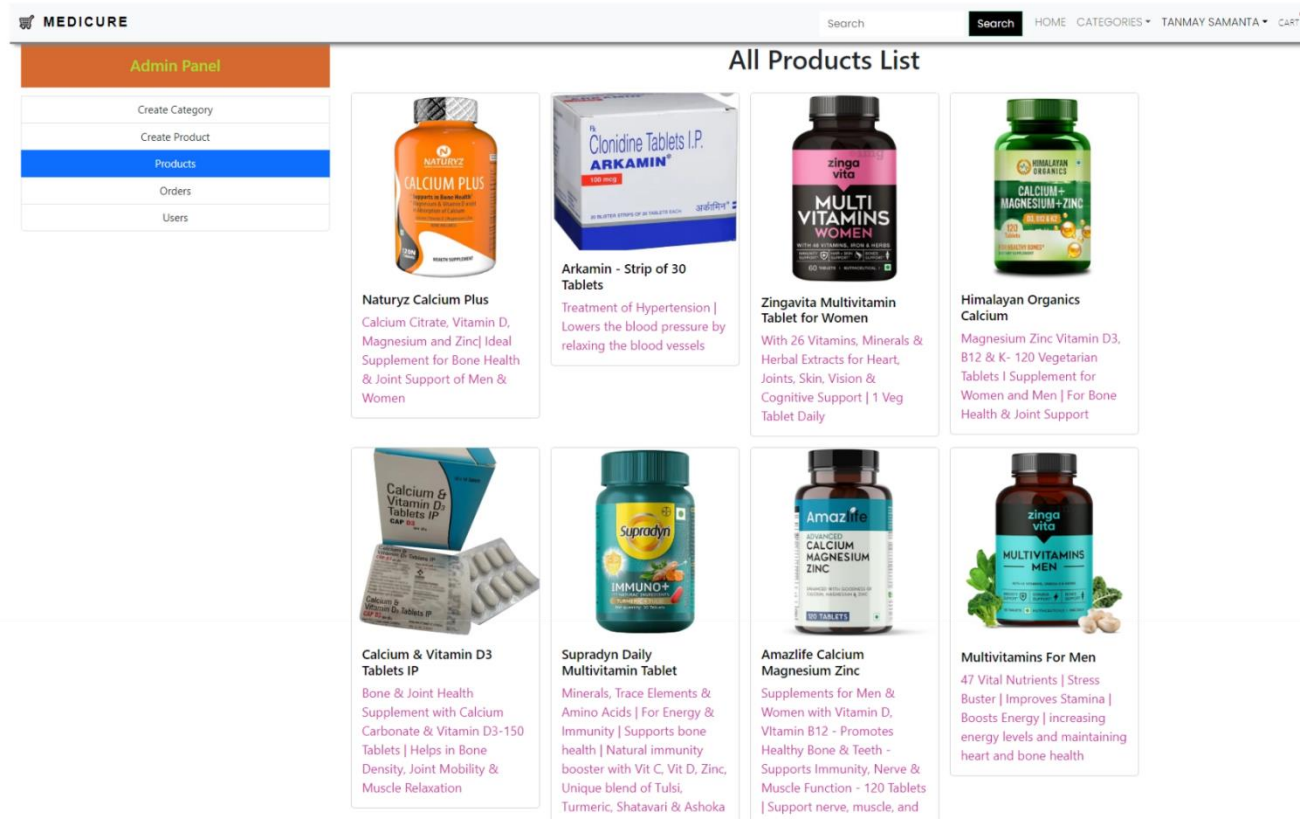
From this page, admin user can create products.

Edit Product Details:



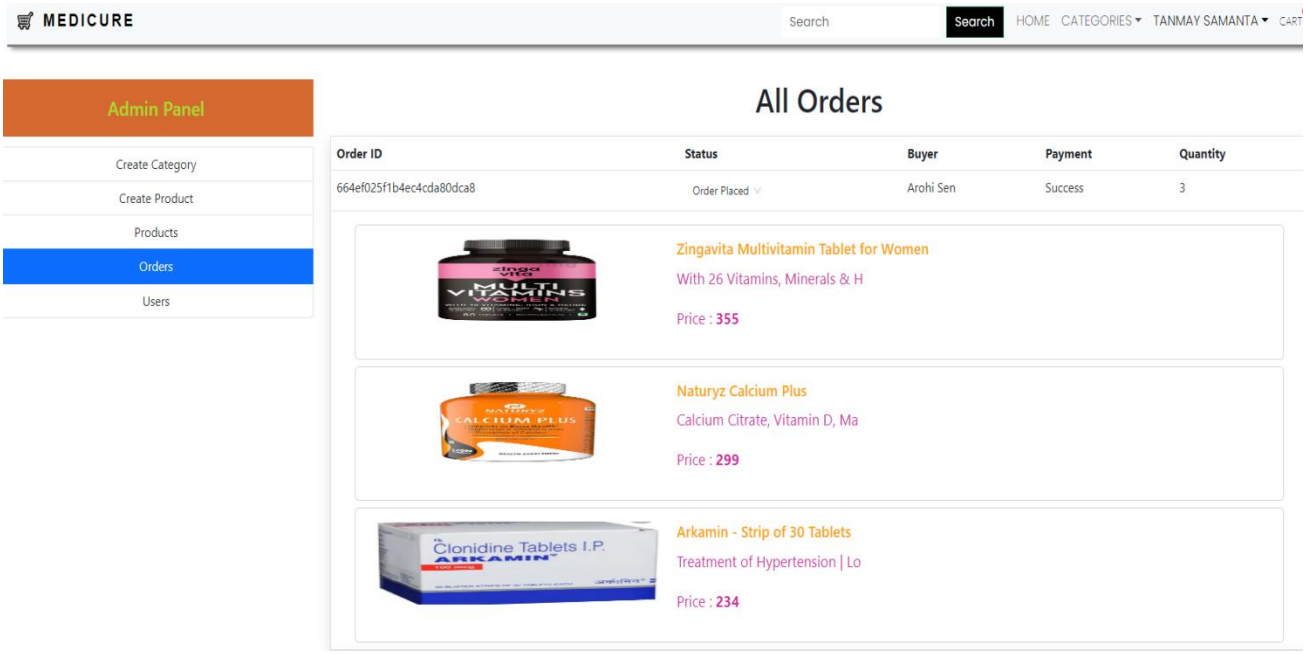
From this page, admin user can edit and delete products along with product details and stock of the products that are available for sale.

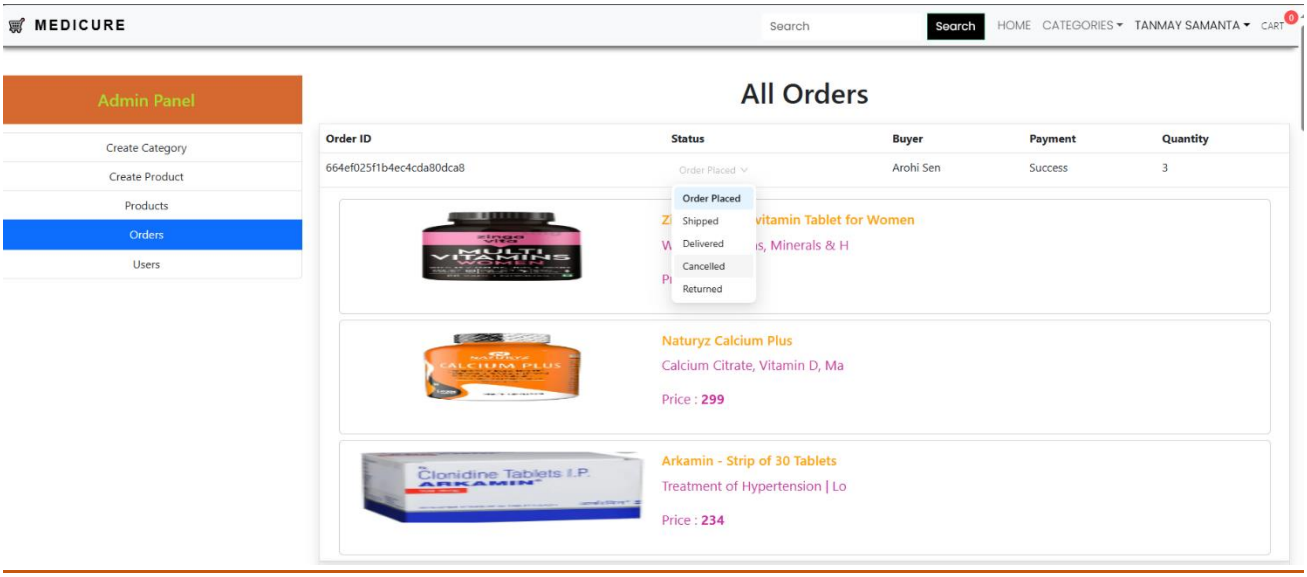
Admin Products Page:



From this page admin can navigate to the required to delete or modify the product.

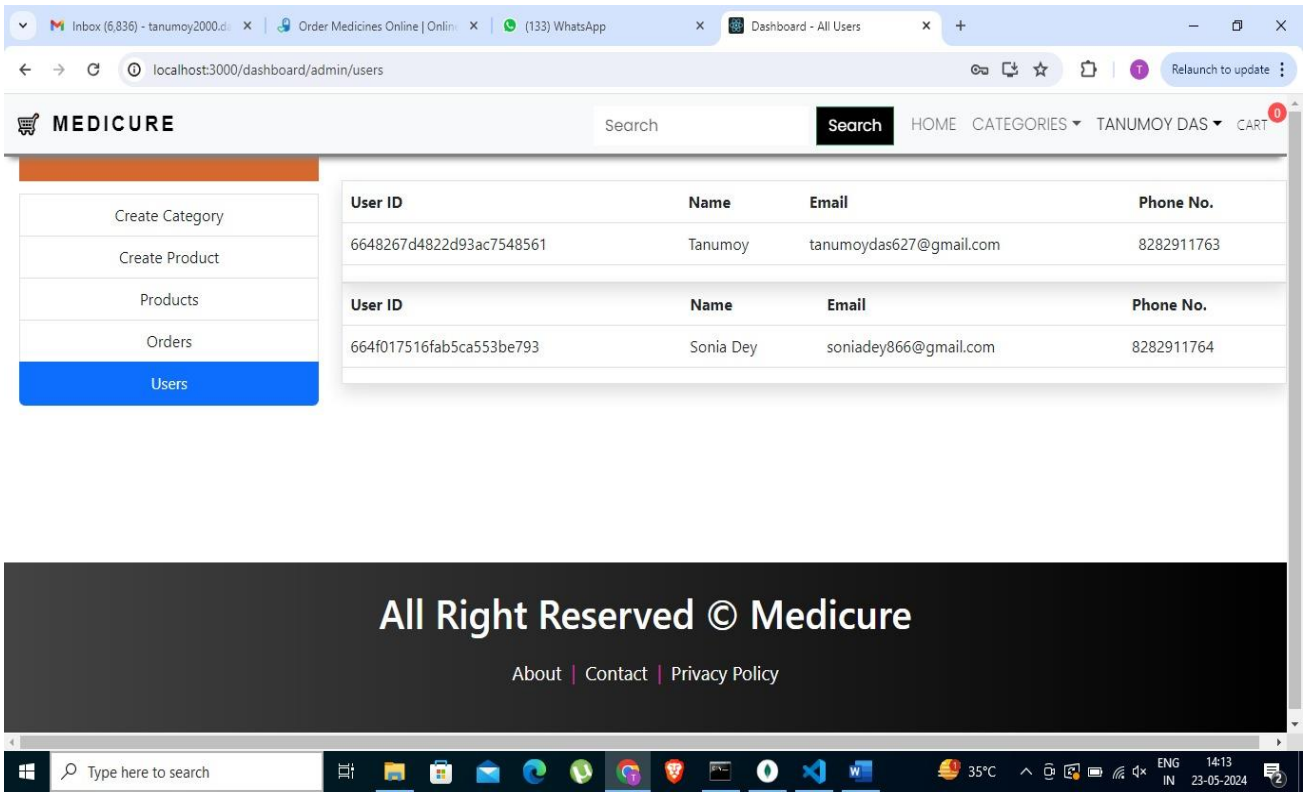
Admin Order Page:





From this page, admin user can view all the orders that are placed and can also update the order status of individual orders by selecting the required option from the "Status" dropdown.

Admin User Page:



From this page, admin user can view user account details.

CONCLUSION

Limitation: Though the software presents a broad range of options to its users, some intricate parts could not be covered into it partly because of logistic and partly due to lack of sophistication. Paucity of time was also a major constraint; thus, it was not possible to make the project foolproof and dynamic.

Considerable efforts have been made to make the software easy to approach even for people who are not related to the field of computers, but it is acknowledged that a layman may find it a bit hard at first instance.

Future Scope: In future, we would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information. We have set our sights on the following additions in future -

- Generating pdf of the order details as an invoice.
- We can give more advanced software including more facilities like auto-reminders for orders.
- We will host the platform on online servers to make it accessible worldwide.
- Display similar items as per the composition.
- Display real-time order tracking.

REFERENCES

We have gathered knowledge about MERN
In the making of this project, we got a lot of help from websites
The sources are: -

- www.node.js
- www.react.dev
- www.mongodb.com
- www.youtube.com
- www.google.com
- https://www.w3schools.com/
- https://www.apachefriends.org/
- https://www.tutorialspoint.com/

Besides, all these sources helped.