CHAPTER - 1

1.1 INTRODUCTION

The aim of Online Medical Store is to automate the existing manual system with the help of computerized equipment and full-fledged computer software and meet the requirements of the customers so that their valuable data/information can be stored for a longer period of time with easy access and manipulation of the same. The required software and hardware is readily available and easy to use. An online medicine warehouse as described above can lead to an error-free, secure, reliable and fast management system.

In this project, users can view the medicines online and place orders at any time. The administrator can add the products and view the order list. The organization can keep computerized records without redundant entries. This means that people are not distracted by information that is not relevant while they are able to access the information. Basically, the project describes how to ensure good performance and better services for customers.

1.2Scope of the capstone project

Problem Statement:

Medical Booking 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

Objectives:

User can order medicine anytime.
Information is private and confidential.
Convenient.
Easy to use, update and maintain.
Time-saving.

Capstone project description:

The purpose of Online Medical Store 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. The required software and hardware are easily available and easy to work with. Online Medical Store, as described above, can lead to error free, secure, reliable and fast management system.

Capstone project Deliverables:

Admin Modules

1. Admin Login -Using user name and password admin can login to this project.
2. Manage Categories - Admin can manage the categories.
3. Manage Products - Admin can manage the products (add and delete).
4. View Orders -Admin can view the list of orders.
5. Offers - In this module admin gives offers to the product.

User Modules

1. User Registration -In this module user can registration with the details.
2. User Login - In this module user can login with his details.
3. View Products - In this module user can view the list of products.
4. Order Products - this module user can send the product orders.
5. Offers - In this module user can view the offers.

Key milestones:

Admin Modules

Admin Login

Manage Categories

Manage Products

View Orders

Offers

User Modules

User Registration

User Login

View Products

Order Products

Offers

Constraints:

No return policy: There are many reasons why drugs may be classified as nonreturnable. The three largest issues include: the product's condition, the timing of the product return relative to the expiration date, and whether the product was originally discounted or sold as nonreturnable by the manufacturer.

Lack of personal and financial security: Every pharmacy website's should publish their privacy policy, one which promises not to share any personal and financial information with third parties. The online financial transactions must be secured through adequate support of encryption technology.

Estimated Capstone project Duration: Estimated capstone project duration is 473hrs.

Estimated Capstone project cost: The total estimated capstone project cost is Rs.44,189

CHAPTER - 2

2.1 CAPSTONE PROJECT PLANNING

2.1.1 Work Breakdown Structure [WBS]

Capstone project Objective(s):

User can order medicine anytime.

Information is private and confidential

Convenient

Easy to use, update and maintain

Time-saving

Work Breakdown Structure – Deliverables

In this project there are two main modules Admin and User.

Admin Modules

a) Admin Login -Using user name and password admin can login to this project.
b) Manage Categories - Admin can manage the categories.
c) Manage Products - Admin can manage the products (add and delete).
d) View Orders -Admin can view the list of orders.
e) Offers - In this module admin gives offers to the product.

User Modules

a) User Registration -In this module user can registration with the details.
b) User Login - In this module user can login with his details.
c) View Products - In this module user can view the list of products.
d) Order Products - this module user can send the product orders.
e) View Offers - In this module user can view the offers.

Activities & Task

a) Admin login (31hrs)

Admin must login with correct details for easy access (task)

Admin must login according to the parameters given for the validation (task)

b) Manage Categories (35hrs)

Admin must manage categories according to products. (task)

Products must be added according to their respective category(task)

c) Manage Products (35hrs)

Admin adds the product to the website. (task)

Admin must manage the products by checking the stock of products (task)

d) View Orders(44hrs)

Admin must view the list of orders (task)

Admin must deliver the products according to the order list(task)

e) Manage Offer's (36hrs)

Admin must manage the offers on the products (task)

User must get offer's on the products. (task)

f) User Registration (42hrs)

User must register using their correct details. (task)

User must register according to the parameters given for the validation. (task)

g) User Login (38hrs)

User must login using their correct details. (task)

User must login according to the parameters given for the validation. (task)

h) View Products (49hrs)

User can view the product's as per their requirements. (task)

The correct description of the product must be displayed according to the products. (task)

i) Order Products (50hrs)

The order list of the products must be displayed according to orders.(task)

Orders must be successful added to the cart(task)

System Architecture Diagram

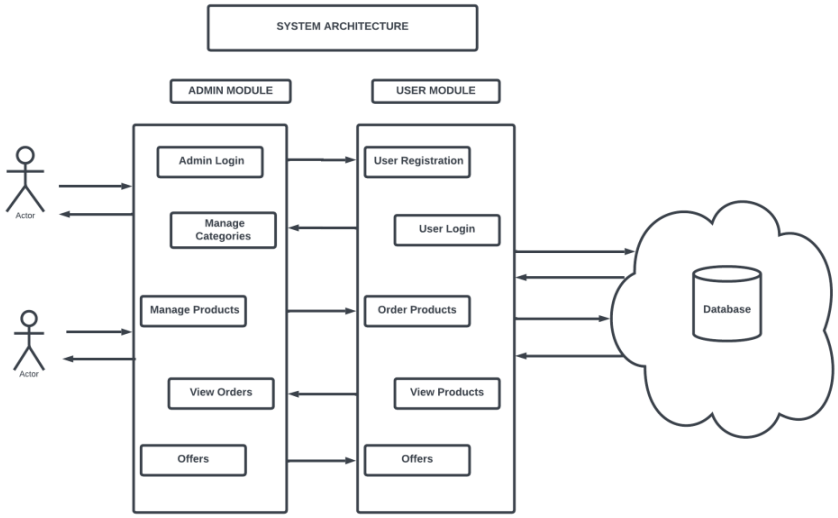

Fig:2.1.1.1

2.1.2 Timeline Development-Schedule

Activities & Task

1. Admin login (31hrs)

   Task 1- Admin must login with correct details for easy access

   Task 2- Admin must login according to the parameters given for the validation.

a) Collect requirement gathering and requirement analysis to create admin login Page.

b) To design admin login form using HTML, CSS, Bootstrap.

c) Validation of user registration form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing admin login form to check validation and database.

✓ This whole task is done by Darshan Ghatge.

2. Manage Categories (35hrs)

Task 1-Admin must manage categories according to products .

Task 2- Products must be added according to their respective category.

a) Collect requirement gathering and requirement analysis to create manage categories page.
b) To design form to manage categories using HTML, CSS, Bootstrap.
c) Validation for managing categories using Javascript.
d) Perform database connection using PHP and Mysql.
e) Testing form to check validation and database.
✓ This whole task is done by Smital Kaginkar.

3.Manage Products (35hrs)

Task 1- Admin adds the product to the website.

Task 2- Admin must manage the products by checking the stock of products .

a) Collect requirement gathering and requirement analysis to create manage products page.
b) To design form to manage products using HTML, CSS, Bootstrap.
c) Validation for manage products form using Javascript.
d) Perform database connection using PHP and Mysql.
e) Testing form to check validation and database.
✓ This whole task is done by Darshan Ghatge.

4.View Orders (44 hrs)

Task 1- Admin must view the list of orders

Task 2- Admin must deliver the products according to the order list.

a) Collect requirement gathering and requirement analysis to create view orders page.
b) To design form to view orders using HTML, CSS, Bootstrap.
c) Validation for to view orders form using Javascript.
d) Perform database connection using PHP and Mysql.
e) Testing form to check validation and database.
✓ This whole task is done by Smital Kaginkar.

5. Manage Offer's (36 hrs)

Task 1- Admin must manage the offers on the products .

Task 2- User must get offer's on the products.

a) Collect requirement gathering and requirement analysis to create manage offers page.

b) To design form to manage offers using HTML, CSS, Bootstrap.

c) Validation for manage offers form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing form to check validation and database.

✓ This whole task is done by Rakshita Jadhav.

6. User Registration (42 hrs)

Task 1- User must register using their correct details.

Task 2- User must register according to the parameters given for the validation.

a) Collect requirement gathering and requirement analysis to create user registration page .

b) To design form to user registration using HTML, CSS, Bootstrap.

c) Validation for user registration form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing form to check validation and database.

✓ This whole task is done by Sufiyan Goundi.

7. User Login (38 hrs)

Task 1- User must login using their correct details.

Task 2- User must login according to the parameters given for the validation.

a) Collect requirement gathering and requirement analysis to create user login page.

b) To design form to user login using HTML, CSS, Bootstrap.

c) Validation for user login form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing form to check validation and database.

✓ This whole task is done by Rakshita Jadhav.

8.View Products (49 hrs)

Task 1- User can view the product's as per their requirements.

Task 2- The correct description of the product must be displayed according to the products.

a) Collect requirement gathering and requirement analysis to create view products page.

b) To design form to view products using HTML, CSS, Bootstrap.

c) Validation for view products form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing form to check validation and database.

✓ This whole task is done by Sufiyan Goundi.

9.Order Products (50 hrs)

Task 1- Orders must be successful added to the cart.

Task 2- The order list of the products must be displayed according to orders.

a) Collect requirement gathering and requirement analysis to create order products page.

b) To design form to order products using HTML, CSS, Bootstrap.

c) Validation for order products form using Javascript.

d) Perform database connection using PHP and Mysql.

e) Testing form to check validation and database.

✓ This whole task is done by Smital Kaginkar.

2.1.3 Cost Breakdown Structure [CBS]

Analyse your Work Breakdown Structure

Admin Modules

Admin Login

Manage Categories

Manage Products

View Orders

Offers

User Modules

User Registration

User Login

View Products

Order Products

Offers

Estimate the labour cost of work:

Module Name	Task Name and their working hours	Total Working Hours	Amount Paid (in hrs)	Total Amount (in ₹)
Admin Login	1. Login with correct details (14hrs) 2.Match the parameters (17hrs)	31Hours	31*70	2170
Manage Categories	1. Manage Categories(18hrs) 2.Add products to the categories(17hrs)	35Hours	35*70	2450
Manage Products	1. Add products(19hrs) 2.Check the products(16hrs)	35Hours	35*70	2450
View Orders	1. View order list(21hrs) 2. Deliver according to the order list(23hrs)	44Hours	44*70	3080
Offers	1. Offer's given on products(17hrs) 2.Offer's received by user(19hrs)	36Hours	36*70	2520
User Registration	1. Register with correct details(20hrs) 2. Match all the parameters(22hrs)	42Hours	42*70	2940
User Login	1. Login with correct details(18hrs) 2.Match the parameters(20hrs)	38Hours	38*70	2940
View Products	1. View products(25hrs) 2.Description of the products(24hrs)	49Hours	49*70	2730
Order Products	1. Display orders(26hrs) 2.Products added to cart(24hrs)	50Hours	50*70	3500

Table:2.1.3.1

Estimate the cost of materials:

Cloud Server Cost: ₹5000/-

Php Designer: $42 (Approx ₹3444/-)

XAMPP Server is open-source software.

Overhead Cost:

If a live server is going to cost more, we will let you know.

Build Contingency into your CBS:

We must complete the job in accordance with the projected cost.

Final-Check:

Estimated Cost:	₹44,189/-
Labour Cost:	₹24,500/-
Material Cost:	₹8,444/-
Profit of the Project: ₹11,245/-	

Table:2.1.3.2

2.1.4 Capstone Project Risks Assessment

he admin login form must function correctly without any errors or problems.The admin login page must display appropriately and have accurate information.

The administrator needs to carefully handle each category.

The administrator needs to carefully manage all of the products in accordance with their classifications.

The administrator should be able to see all orders.

The user should register using the proper information and specifications.

User information should validate without any problems.

The user login page responded as expected.

In order to purchase their product, the user needs view the order list.

2.2 REQUIREMENT SPECIFICATION

2.2.1 Functional

Admin

a) Login to the system.
b) Manage the categories
c) Add or remove the categories
d) Manage the products
e) View the orders
f) Handle the offers

User

a) Requesting registration
b) Login to the system
c) View the product list
d) Order the products
e) Use the coupon for offers.

2.2.2 Non-Functional(Quality Attributes)

Admin

a) User name
b) Password

Category

a) Category name

Orders

a) Product id
b) User id
c) Address
d) City
e) State
f) Zip code

g) Payment method
h) Notes
i) Total
j) Create date
k) Status

Products

a) Product name
b) Product code
c) Product description
d) Image
e) Category id
f) Cost

User

a) Username
b) Password
c) Name
d) Email
e) Mail

User coupon

a) Coupon code
b) Discount
c) User id
d) Created data
e) Status

2.2.3 User inputs

Admin

a) User name and password
b) Manage Categories
c) Manage Products
d) Manage offers

User

a) Full name, User name ,Password, Email id,
b) Contact number ,Select year ETC
c) User name and password
d) View the products
e) Order Products
f) Use the coupons

2.2.4 Technical Constraints

Software requirement for deployment:-

a) OS.
b) Chrome.

Language used or technology:-

a) Content language (client side language) (HTML, CSS, bootstrap, JS).
b) Backend (server side language) (PHP).
c) Database (My SQL).

2.3 DESIGN SPECIFICATION

2.3.1 Chosen System design

System architecture diagrams offer a visual representation of a system's numerous components and demonstrate how they speak to one another and interact. The architecture and structure of a system are described in these diagrams. Identification of requirements that have an impact on the application's structure is the architecture's main objective. How the modules and database communicate with one another is depicted in the figure below. We have three primary modules, including an admin module and a user module.

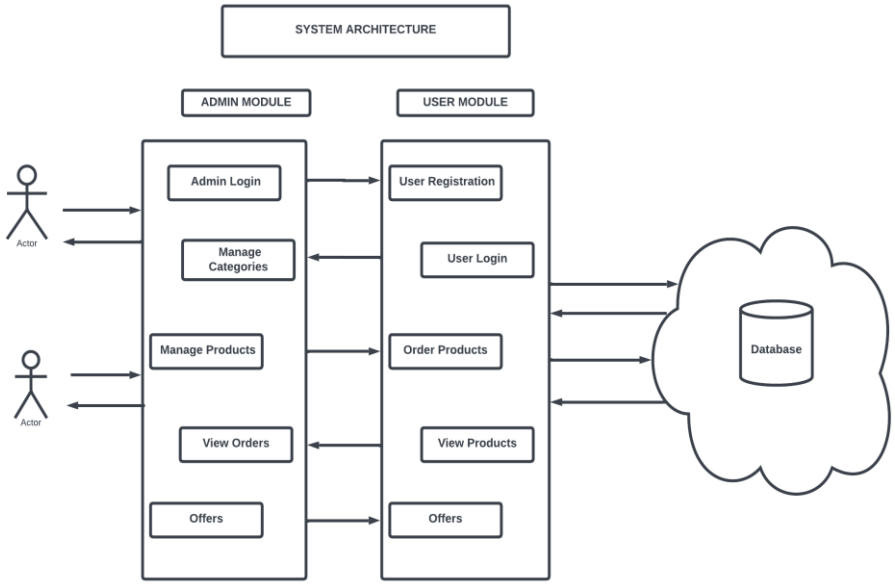

Fig:2.3.1.1

2.3.2 Detailed of Alternative Designs

A data flow diagram (DFD) is a graphical or visual depiction that describes how data is moved through an organization's activities using a standardised set of symbols and notations. The path that data follows between external entities, processes, and data repositories is known as data flow. Arrows are used to depict the interaction between the other components, and they are usually labelled with a brief data name, such as "E-commerce Online Medicine Shopping".

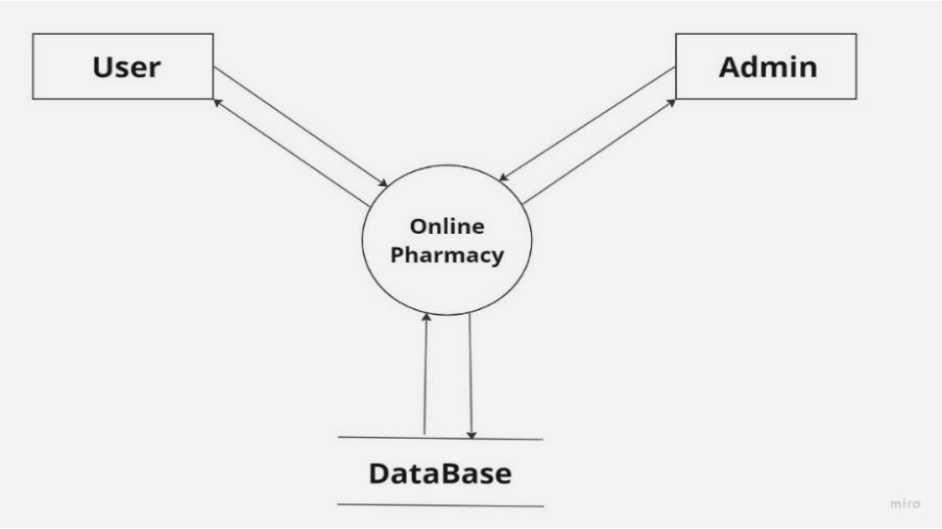

Fig:2.3.2.1

DATA FLOW DAIGRAM LEVEL -1[Admin]

A data flow diagram (DFD) is a graphical or visual depiction of a business's processes using a standardised set of symbols and notations. The path that data follows between external entities, processes, and data repositories is known as data flow. It depicts the interaction with the other components and is represented by arrows, which are usually labelled with a brief data name, such as "Admin Login."

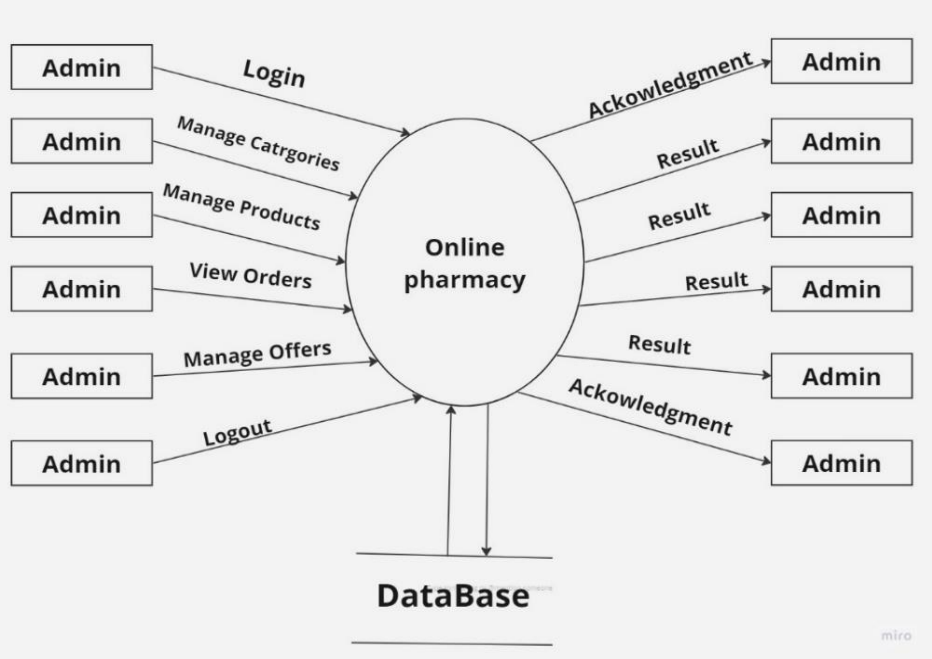

Fig:2.3.2.2

DATA FLOW DAIGRAM LEVEL -1[User]

A data flow diagram (DFD) is a graphical or visual depiction of a business's activities through data movement that uses a standardised set of symbols and notations. Data flow is the path that information travels as it moves between external entities, processes, and data repositories. It represents the interface with the other elements and is represented by arrows, which are often labelled with a brief data name, such as “User Login”.

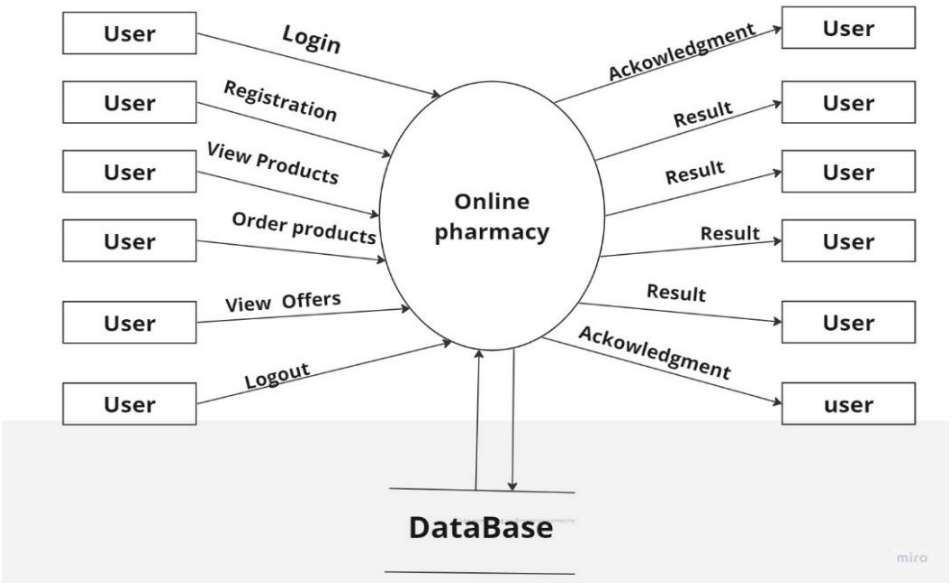

Fig:2.3.2.3

2.3. Detailed Description of Components/Subsystems

Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system. A component diagram is used to illustrate the connections between various system components. Component diagrams display the organisation of the software system, which includes information on the interfaces and dependencies of the programme components. Here are the elements. Here, we have displayed the module dependencies in this diagram

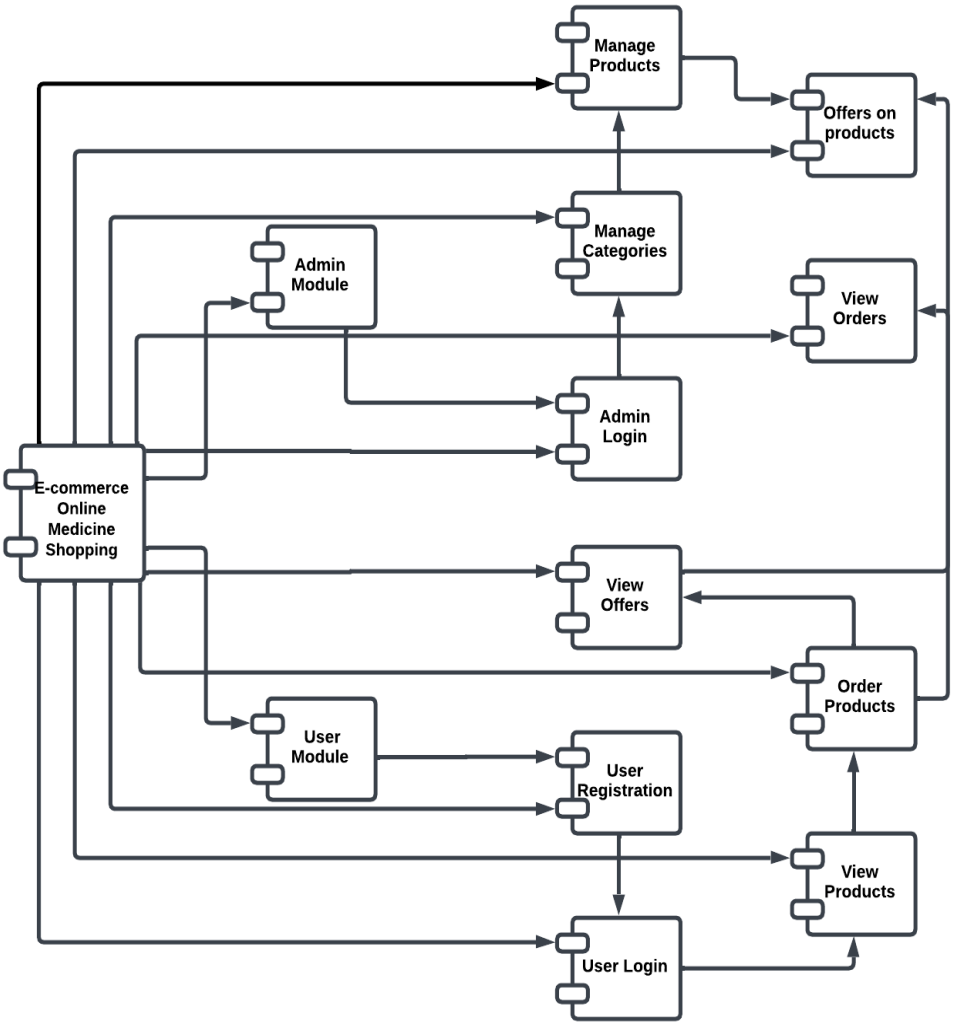

Fig:2.3.3.1

2.3.4 Component 1-n

1. Admin Login:

Admin must login with correct details. Admin must have correct username and password details that matches the parameters. Admin manages whole of the process from adding categories, products to the appropriate category, product details and images, offers on the products, viewing the order list.

2. Manage Categories:

Categories are according to admin's requirement and are managed by admin itself. There can be number of different categories where products can be viewed according to category they belong.

3. Manage Products:

The products are added to the respective categories .Here the products are displayed with their correct names, description along with image.

4. View Orders:

Here admin can view the orders which are ordered by the users(customer's).

5. Offers:

There will be offer's on the products where users get offers on the products they ordered where a coupon code is generated using the code the discount is given on the order.

6. User Registration:

Here user must register with correct details like name, username, password, email, phone-number, and address where user gets username and password to login

7. User Login:

User can login using this correct details as he gets a username and password above for easy access to order products.

8. View Products:

Here user views the products with their image, name, description, or according to their requirements and then can order products.

9. Order Products:

Here user orders the product they need and gets offer on the products they order for the next time.

CHAPTER-3

3.1 APPROACH AND METHODOLOGY

3.1.1 Discuss the Technology

Web Technology:

Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the Internet. A web browser is used to access web pages. Web browsers can be defined as programs that display text, data, pictures, animation, and video on the Internet.

→ Types of Web Technology:

a) Browsers
b) HTML and CSS
c) Programming Languages
d) Frameworks
e) Web Servers
f) Databases Protocols
g) Data Formats

→ Advantages of web technology.

a) Independently accessed from any location.
b) There is no data loss, we can recover the data.
c) Save the time.
d) Easy Data Sharing and Collaboration.
e) Centralized Security.

Cloud Based Technology:

To live the project, we need to purchase cloud server.

Cloud computing technology gives users access to storage, files, software, and servers through their internet-connected devices: computers, smartphones, tablets, and wearables. Cloud computing providers store and process data in a location that's separate from end users.

→ Types of Cloud Based technology:

a) SAAS:- Software as a Service.

b) PAAS:- Platform as a Service.

c) IAAS:- Infrastructure as a Service.

→ Advantages of Cloud Based Technology:

a) Accessibility anywhere, with any device

b) Ability to get rid of most or all hardware and software

c) Higher performance and availability

d) Price-performance and cost savings.

Open Source Technology:

For our project we not require to purchase any software's and libraries.

a) XAMPP

Xampp is a cross-platform and open-source tool, which makes it an idea choice of web developers. It is the acronym of X-Cross Platform, Apache, Mysql, PHP, Perl.

b) PHP

PHP is an open-source scripting language used for creating dynamic and interactive web pages and various digital platforms.

c) PhpMyAdmin

Php Admin is an open source and free administration tool for mysql.

d) Software Libraries:

Software library are freely available.

3.1.2 Methodologies

Agile Methodology is a process for managing a project that involves constant collaboration & working in iteration.

Agile project Management work off the basis that a project can be continuously improved upon thought its life cycle, with changes being made quickly & responsibility.

Phases Of Agile Model:-

a) Requirement Gathering.
b) Design the requirement.
c) Construction/Iterations.
d) Testing/Quality Assurances.
e) Deployment.
f) Feedback.

Agile Diagram:-

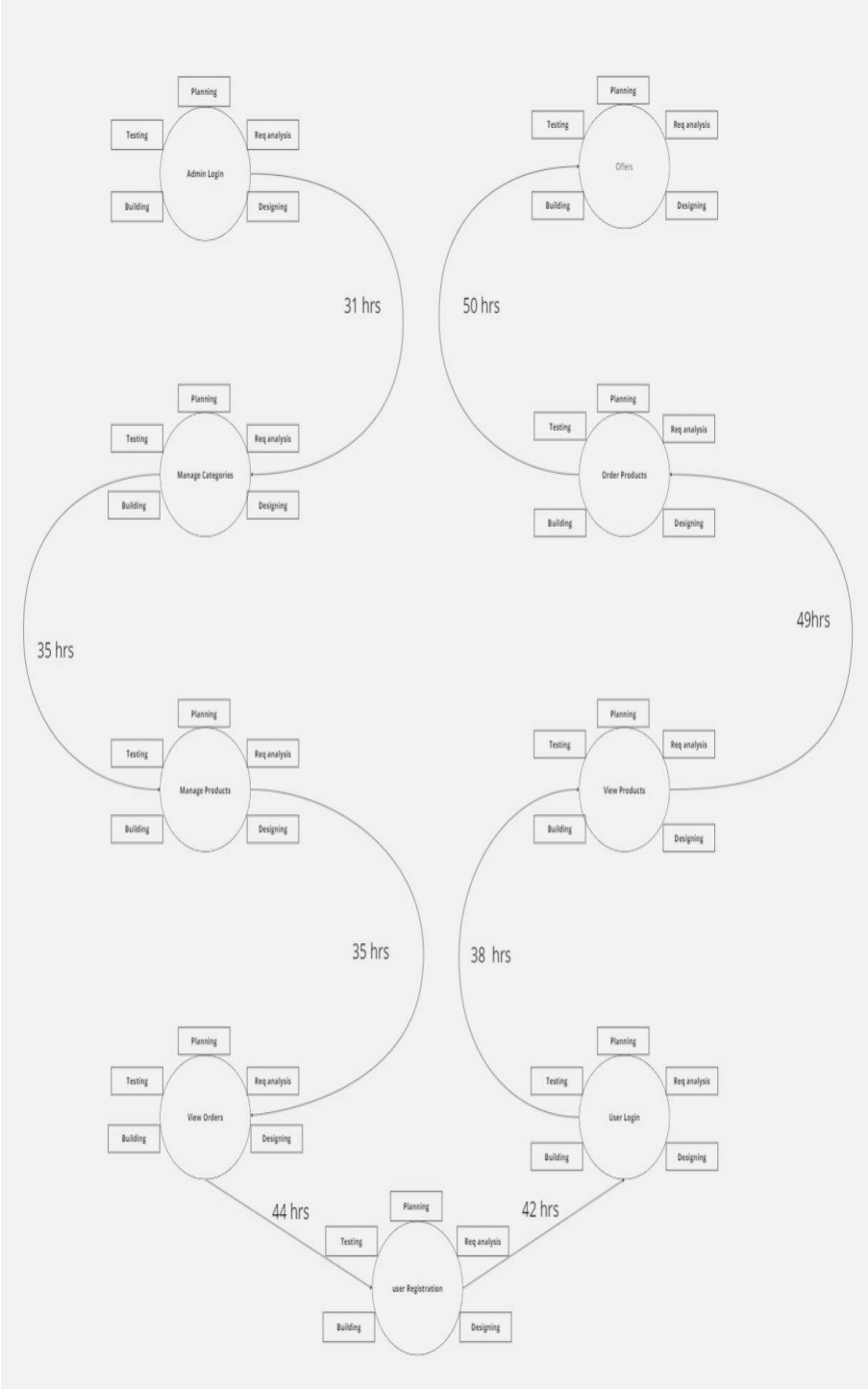

Fig:3.1.2.1

3.1.3Use Case

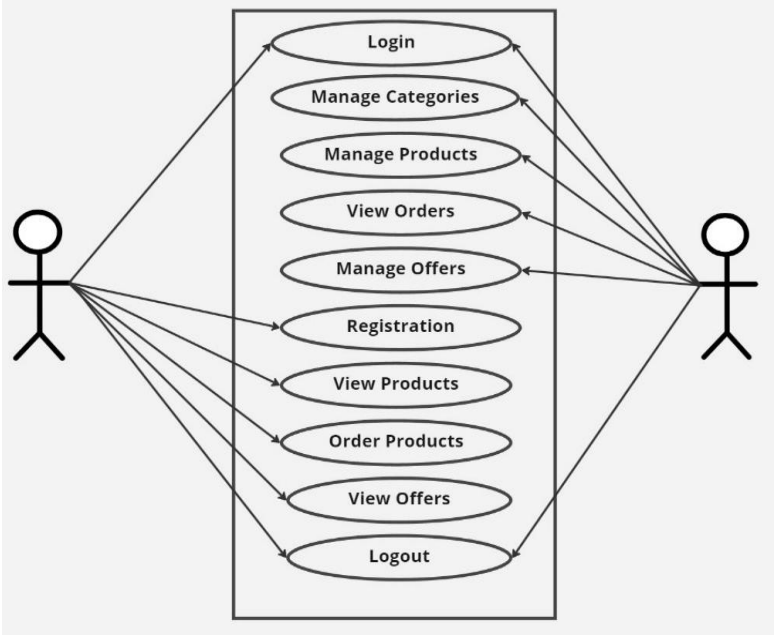

Fig:3.1.3.1

3.1.4 Programming

a) HTML:

HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. Is a client-side scripting language. Hypertext defines the link between the web pages and markup language defines the text document within the tag that define the structure of web pages.

b) CSS:

CSS stands for Cascading Style Sheets.CSS describes how HTML elements are to be displayed on screen, paper, or in other media.CSS saves a lot of work. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

c) JavaScript:

JavaScript (JS) is the most popular lightweight, interpreted compiled programming language. It can be used for both **Client-side** as well as **Server-side** developments. JavaScript also known as a scripting language for web pages.

d) Bootstrap:

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML,CSS and JS framework for developing responsive, mobile-first websites. It is Faster and Easier way for Web-Development. It creates Platform-independent web-pages. It creates Responsive Web-pages.

e) PHP:

PHP is an acronym for "PHP: Hypertext Pre-processor". PHP is a widely-used, open source scripting language. PHP scripts are executed on the server. PHP is free to download and use.

3.1.5 Analysis

a)Existing System: -

The existing medical Booking 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.

b)Proposed System: -

The main advantage of the proposed the system is the user can easily find out the availability of particular medicine through own devices. This system also shows the nearest pharmaceutical shops on their devices. This system can be divided into various sections.

In this project user can view the medicines online and give the orders any time. Admin can add the products and view the order list. The proposed medical Booking Store system will completely Revolutionize the industry. Searching of products, order placing, billing and product stock can be maintained by a single click.

3.1.6 Process design

Databases Tables:

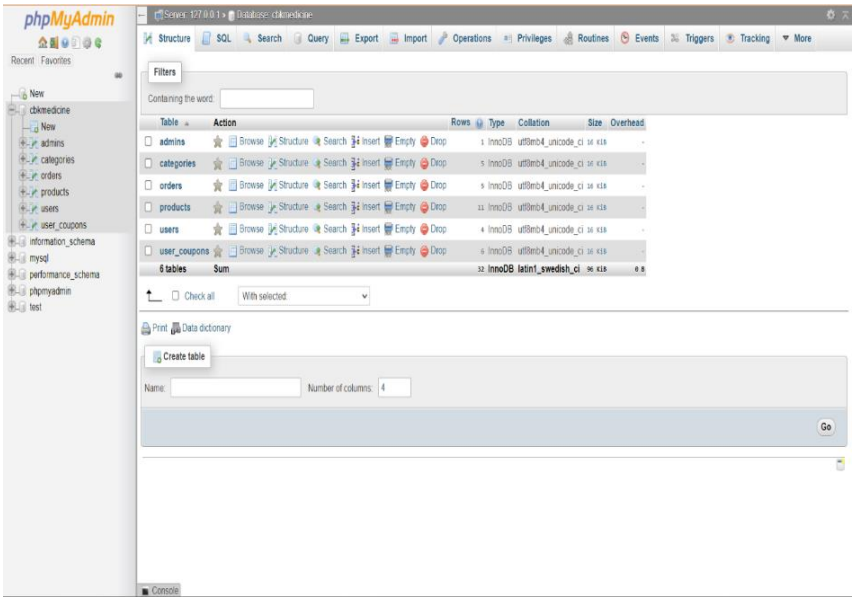

Fig:3.1.6.1

Admin Login:

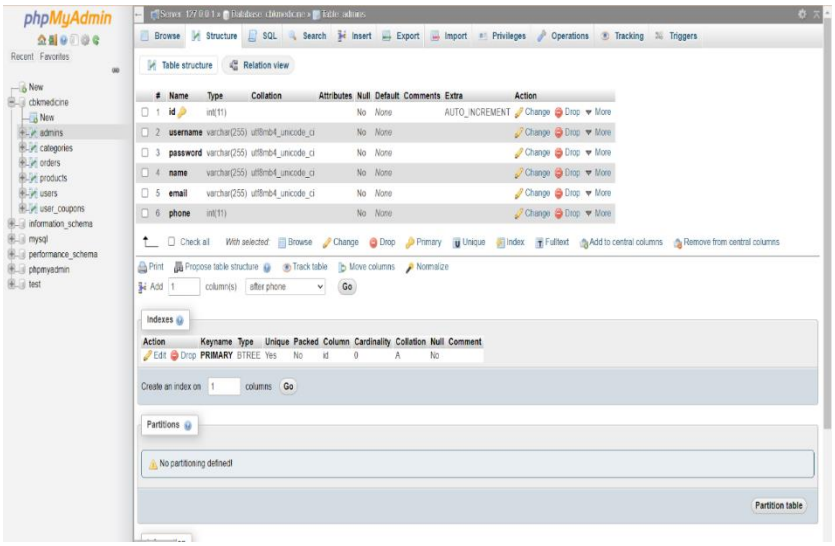

Fig:3.1.6.2

Manage Categories:

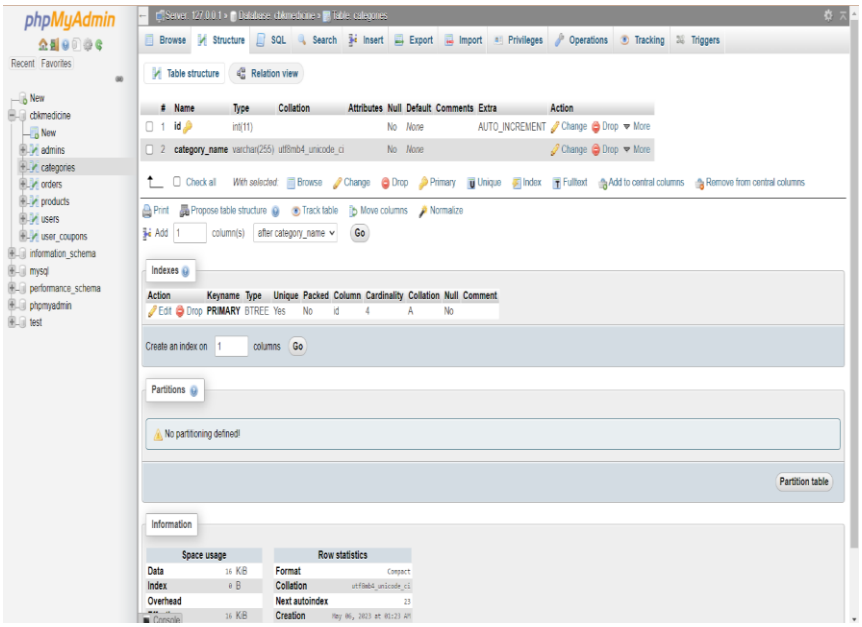

Fig:3.1.6.3

Manage Products:

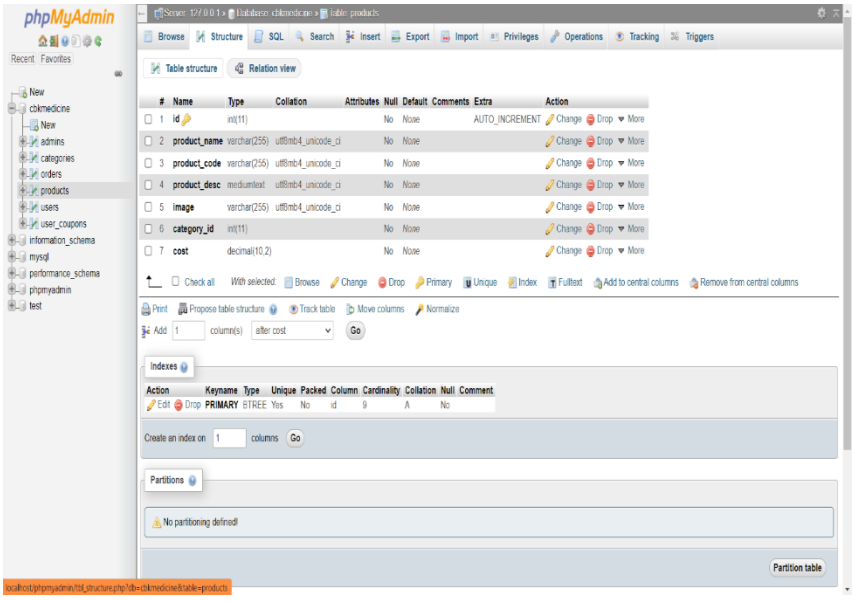

Fig:3.1.6.4

Ordered products:

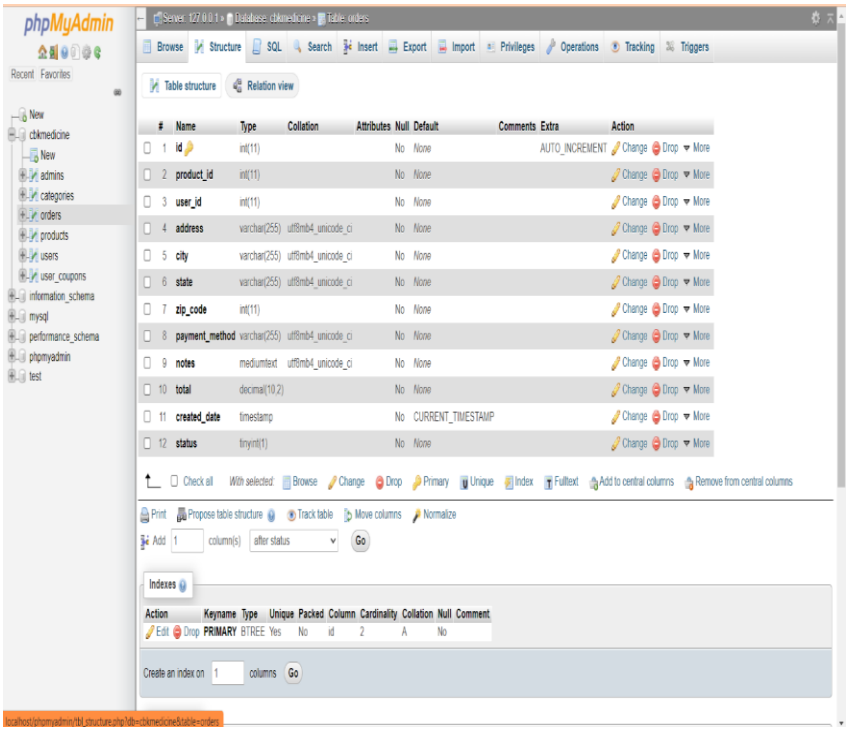

Fig:3.1.6.5

User Login:

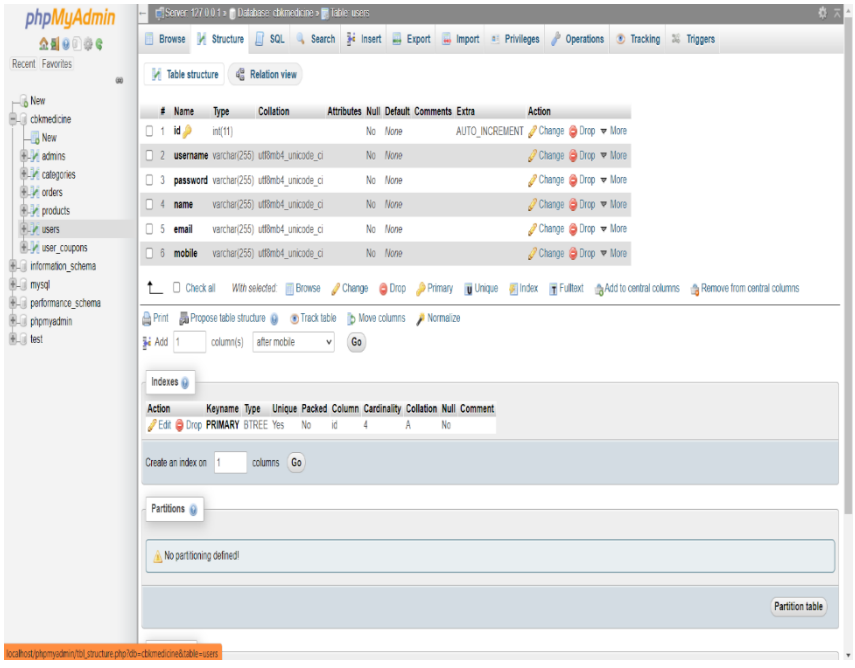

Fig:3.1.6.6

Offers:

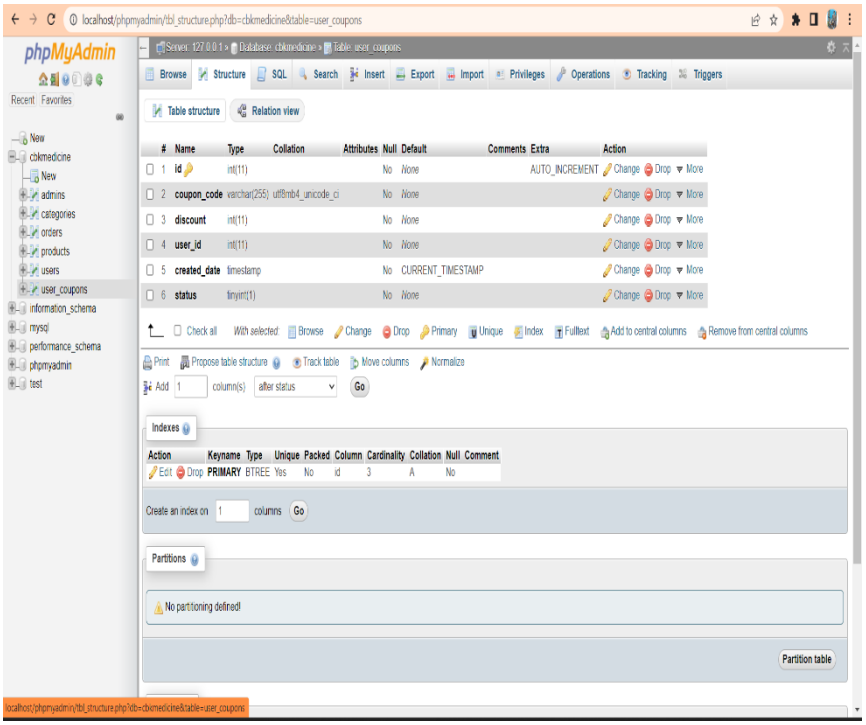

Fig:3.1.6.7

3.1.7 Product design

Admin Login

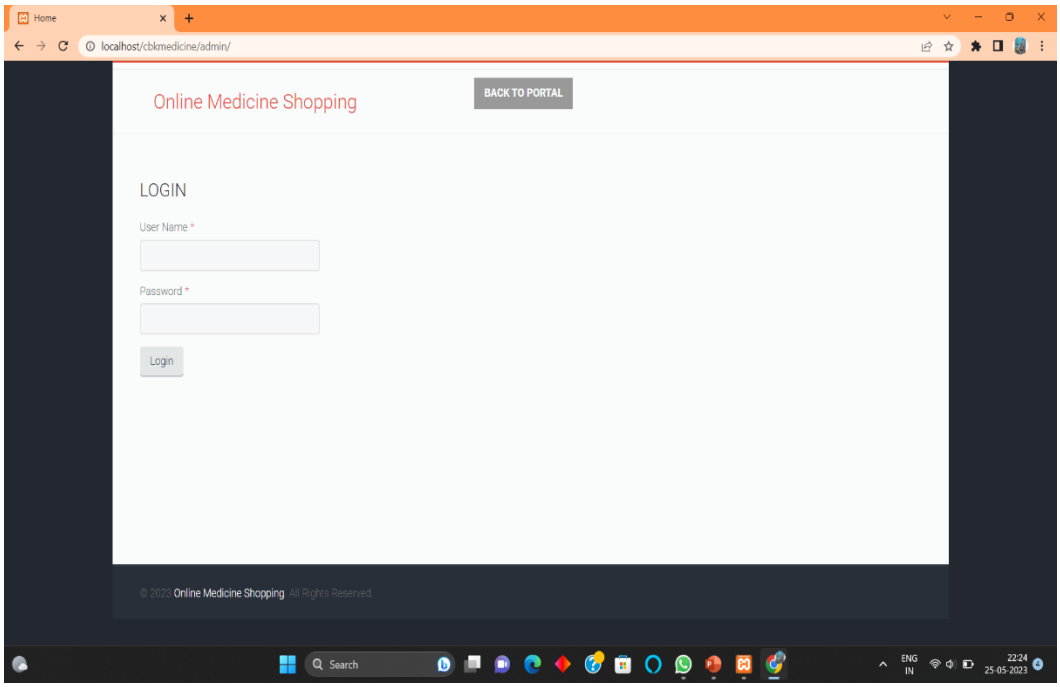

Fig:3.1.7.1

Add Category

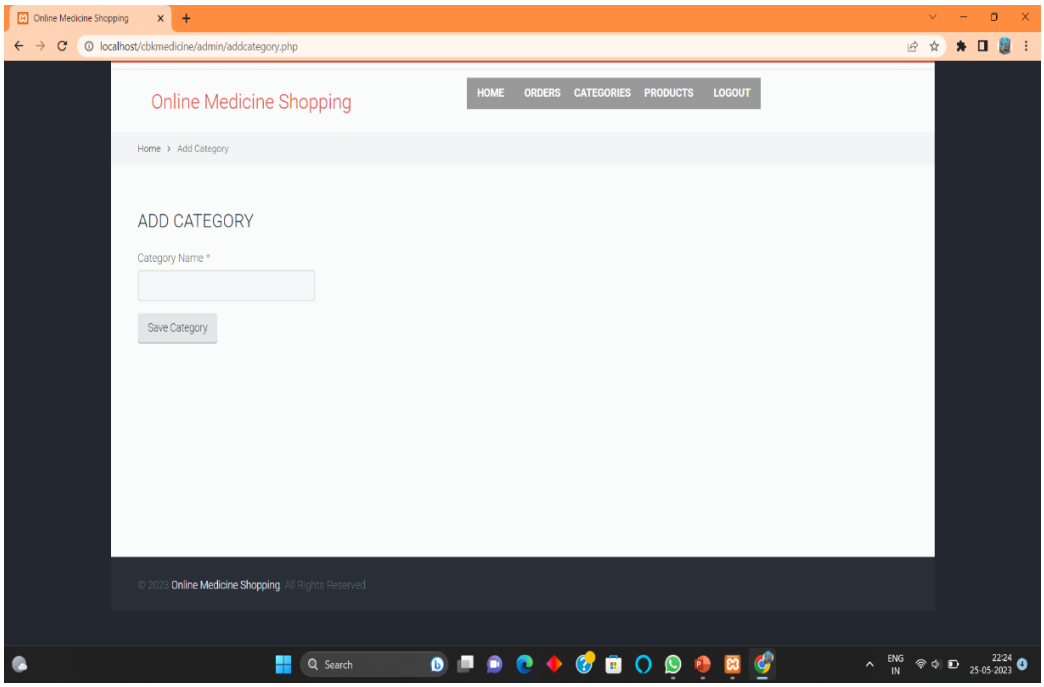

Fig:3.1.7.2

Add Product

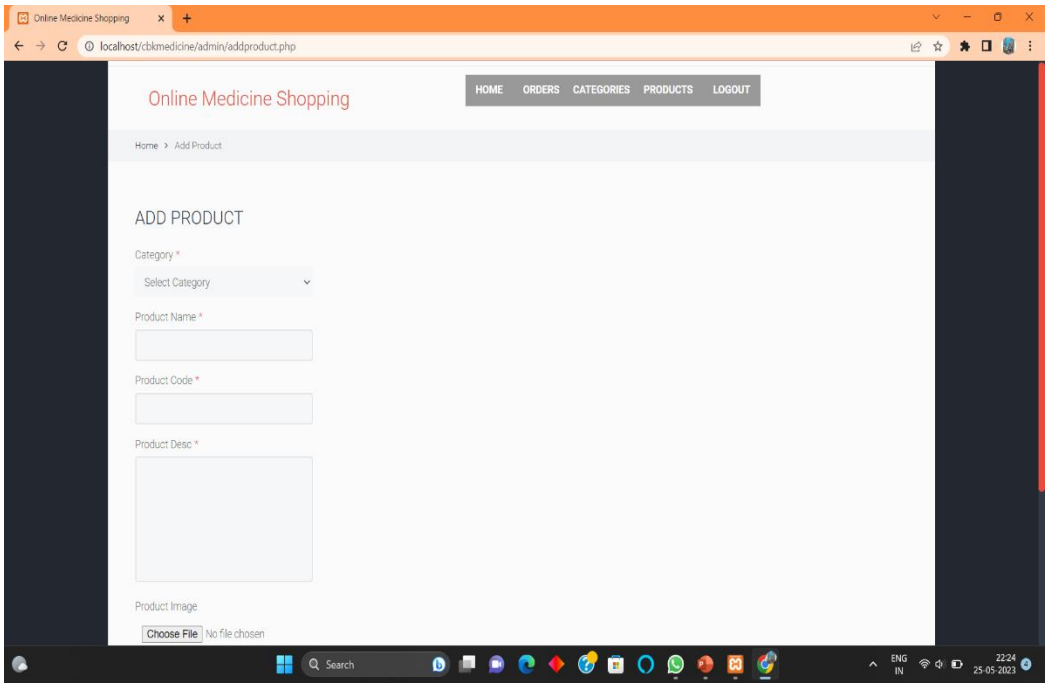

Fig:3.1.7.3

View Orders

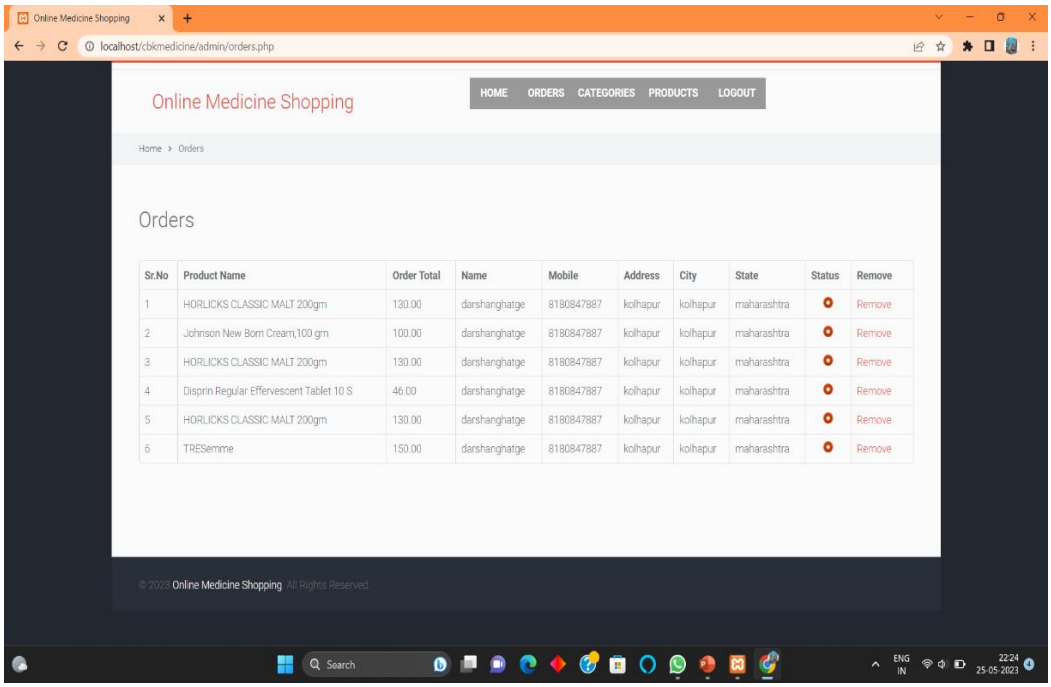

Fig:3.1.7.4

User Registration

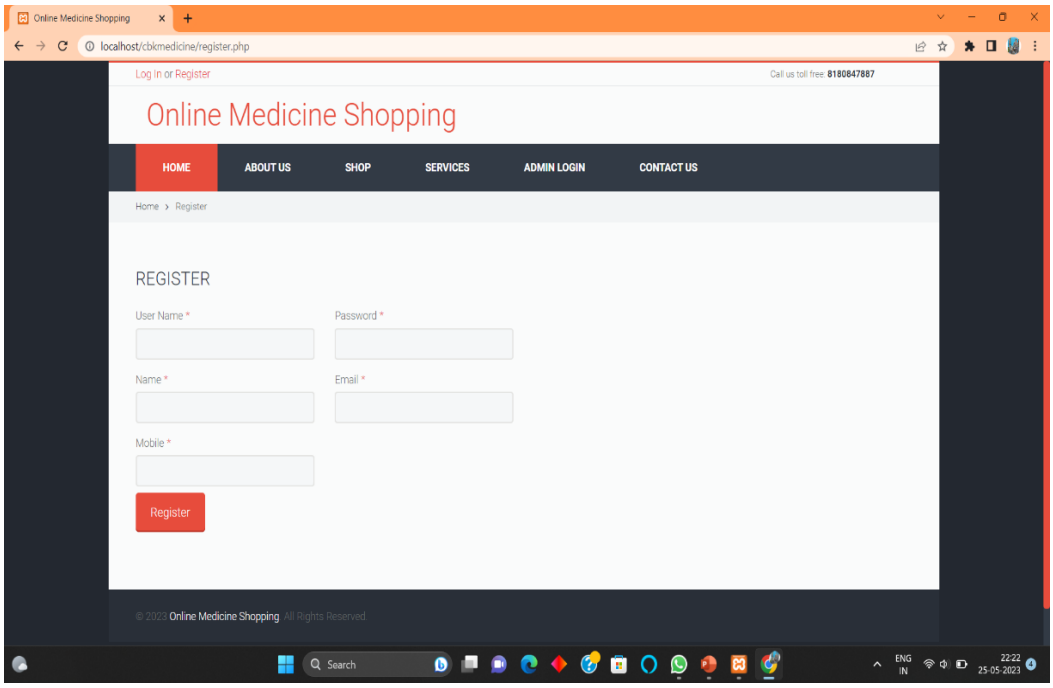

Fig:3.1.7.5

User Login

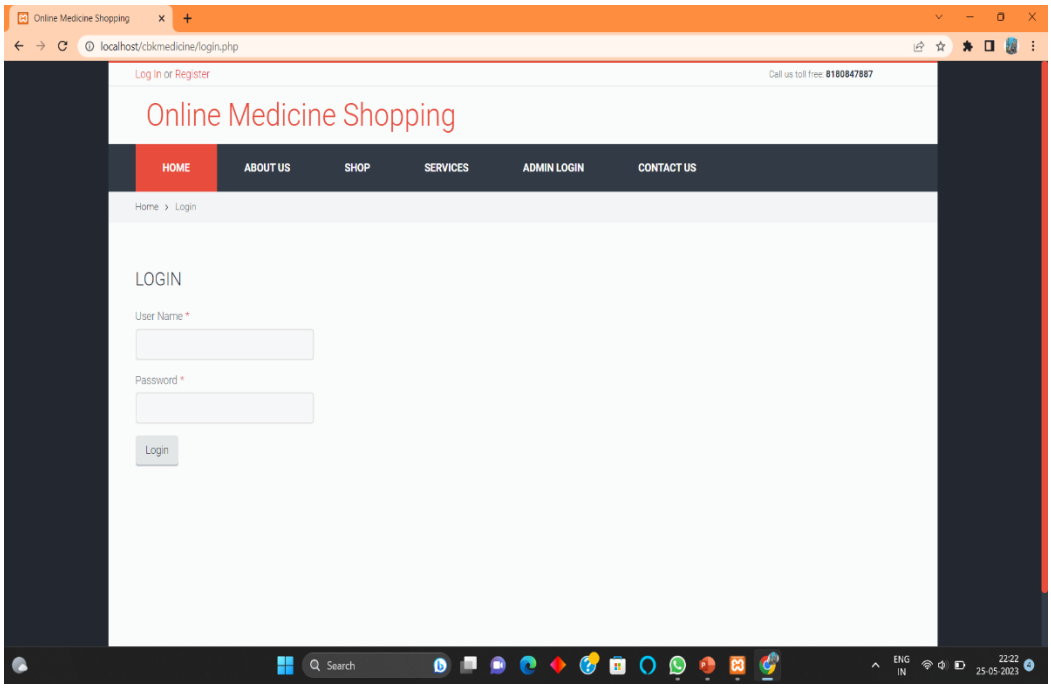

Fig:3.1.7.6

View products

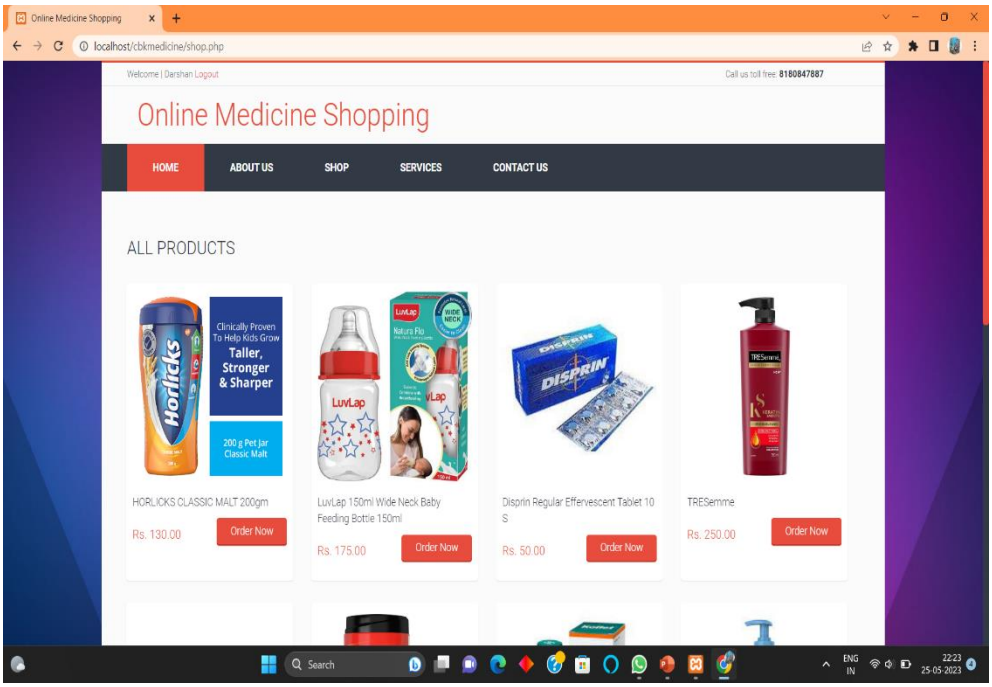

Fig:3.1.7.7

Order products

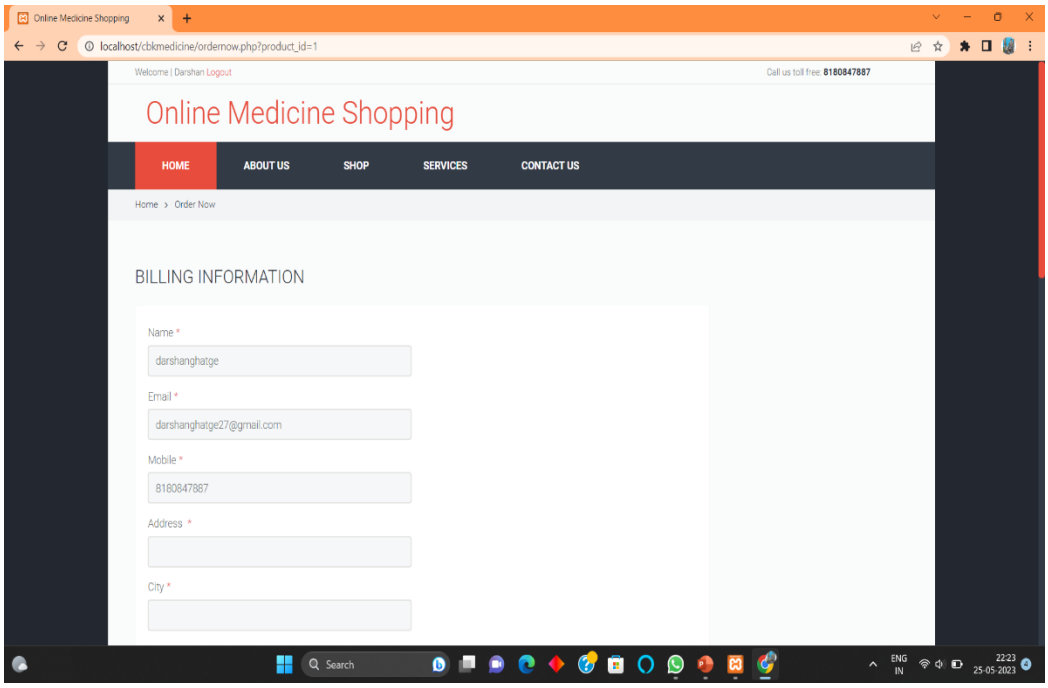

Fig:3.1.7.8

Offer's

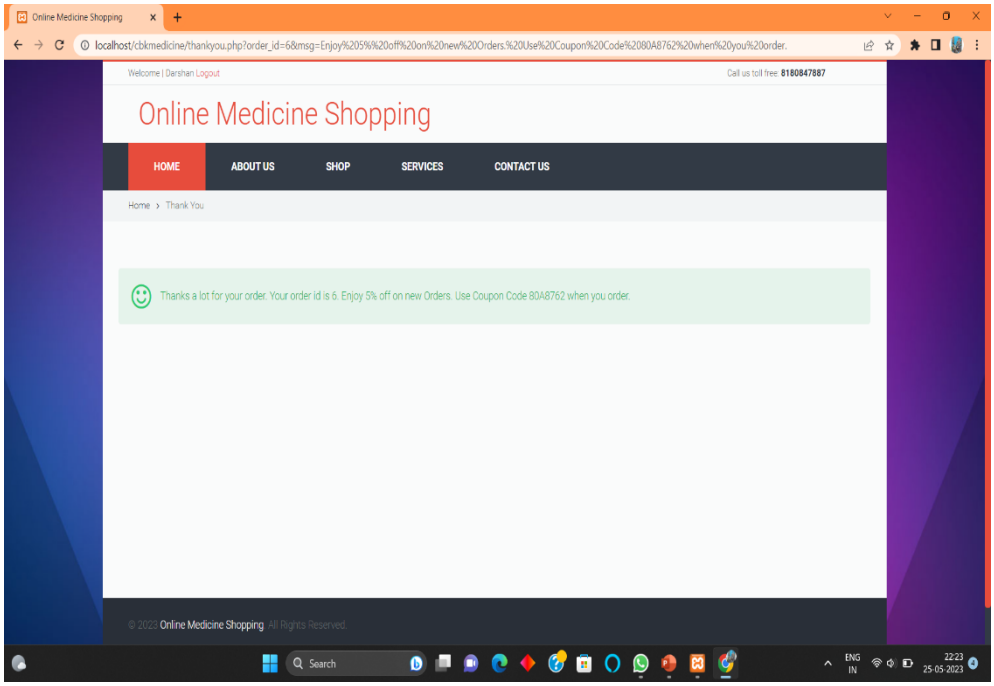

Fig:3.1.7.9

3.1.8 Fabrication

Admin login:

In admin login, first we have collected the requirements then analyse the requirements gathered. Then we design the admin login form. After designing the page it is validated, where the form must be filled with correct details in each field . Then we performed the database connection. And finally, after completing all the steps testing is done.

Manage Categories:

For managing categories, first we have collected the requirements then analyse the requirements gathered. Then we design the add categories page. After designing the page it is validated, where the categories are specified to match the correct parameters. Then we performed the database connection. And finally, after completing all the steps testing is done.

Manage Products:

For managing products, first we have collected the requirements then analyse the requirements gathered. Then we design the page to add products. After designing the page it is validated, where the products must be added in specific category it belongs to. Then we performed the database connection. And finally, after completing all the steps testing is done.

View Orders:

To view orders, first we have collected the requirements then analyse the requirements gathered. Then we design view order page. After designing the page it is validated, where order list is displayed with the list of products and user details. Then we performed the database connection. And finally, after completing all the steps testing is done.

Offers:

For offers, first we have collected the requirements then analyse the requirements gathered. Then we design the offer page where coupon code is generated. After designing the page it is validated, where after every order the coupon code is generated. Then we performed the database connection. And finally, after completing all the steps testing is done.

User Registration:

For user registration, first we have collected the requirements then analyse the requirements gathered. Then we design user registration page according to requirements. After designing the page it is validated, where user must register with correct details and match the specifications. Then we performed the database connection. And finally, after completing all the steps testing is done.

User Login:

For user login, first we have collected the requirements then analyse the requirements gathered. Then we design user login page according to requirements. After designing the page it is validated, where user must login with correct details and match the specifications. Then we performed the database connection. And finally, after completing all the steps testing is done.

View Products:

To view products, first we have collected the requirements then analyse the requirements gathered. Then we design product page according to requirements. After designing the page it is validated, where user can view the products given by admin with correct product details. Then we performed the database connection. And finally, after completing all the steps testing is done.

Order Products:

To order product, first we have collected the requirements then analyse the requirements gathered. Then we design order page according to requirements. After designing the page it is validated, where user orders product and gets a coupon code. Then we performed the database connection. And finally, after completing all the steps testing is done.

CHAPTER-4

4.1 TEST AND VALIDATION

4.1.1Test Plan:

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

Software testing can be divided into two steps:

a) Verification: it refers to the set of tasks that ensure that the software correctly implements a specific function.
b) Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Testing Types: -

Manual Testing: -
Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

Unit Testing: -
Unit testing is a software development process in which the smallest testable parts of an application, are called units. Unit testing is an important step in the development process. If done correctly, unit tests can detect early flaws in code which may be more difficult to find in later testing stages.

Integration Testing: -

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

White Box Testing: -

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

Black Box Testing: -

The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

4.1.2 Test Approach:

Validation:

Admin Login

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC01	Check Admin Login with Valid Data	Enter Admin name And Password	Admin name = admin Password = admin	Admin Login Successfully	As Expected	Pass

Table:4.1.2.1

Manage Categories

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC02	Check Admin has added category with valid data	Enter Category name	Category Name= Health and nutrition	Category add successfully	As Expected	Pass

Table:4.1.2.2

Manage Product's

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC03	Check Admin adds products with their correct details	Product Name, Product code, Description etc	Product Name = Horlicks Classic Malt 200g	Product add successfully	As Expected	Pass

Table:4.1.2.3

View Order

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC04	If admin doesn't want to take order admin can remove it	Orders	Remove	Order removed successfully	As Expected	Pass

Table:4.1.2.4

User Registration

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC05	Check User Register with Valid Data	Enter User Name , Name , Mobile number, Password , Email address.	User Name= Darshan, Name= Darshan , Mobile number= 880847887, Password= 123456, Email address= darshanghatge@gmail.com	You Success-fully Registered	As Expected	Pass

Table:4.1.2.5

User Login

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC06	Check User Login with Valid Data	Enter Username And Password	Username = admin Password = admin	User Login Successfully	As Expected	Pass

Table:4.1.2.6

Order Product's

Test Case Id	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status
TC04	Check user gives appropriate details for	Enter Address ,City , State, Zip code etc	Address=Kolhapur ,City=Kolhapur , State=Maharashtra, Zip code=416006 etc	Thanks a lot for your order	As Expected	Pass

Table:4.1.2.7

4.1.3 Features Tested:

Modules

a) Admin Login
b) Manage Categories
c) Manage Products
d) View Orders
e) Offers

User Modules

a) User Registration
b) User Login
c) View Products
d) Order Products
e) Offers

4.1.4 Features Not Tested:

We have not tested on cloud server because we need to purchase cloud server.

4.1.5 Findings:

Admin login

a) Empty Username: - If admin doesn't provide any username in username field, we will receive a pop-up message as please enter username, and we expected the same result, hence the test is passed.

b) Empty Password: - If admin doesn't provide any password in password field, we will receive a pop-up message as please enter password, and we expected the same result, hence the test is passed.

Manage Categories

a) Empty Category Name: - If admin doesn't provide any Category name in add categories field, we will receive a pop-up message as please enter Category name, and we expected the same result, hence the test is passed.

b) Update Categories: - If admin wants to Update the categories then admin will Edit it and update it , then hi he will receive a pop-up message as category successful update.

Manage Product

a) Empty Product Name: - If admin doesn't provide any product name in add product field, we will receive a pop-up message as please enter product name, and we expected the same result, hence the test is passed.

b) Product Details: - If admin doesn't provide any product detail's in product detail's field, we will receive a pop-up message as please enter product detail's, and we expected the same result, hence the test is passed.

View Orders

a) Remove Order: - If admin doesn't provide any product name in add product field, we will receive a pop-up message as please enter product name, and we expected the same result, hence the test is passed.

User Registration

a) Empty Username: - If user doesn't provide any username in username field, we will receive a pop-up message as please enter username, and we expected the same result, hence the test is passed.

b) Empty Email:- If user doesn't provide any email in email field, we will receive a pop-up message as please enter email address, and we expected the same result, hence the test is passed.

User Login

a) Empty Username: - If user doesn't provide any username in username field, we will receive a pop-up message as please enter username, and we expected the same result, hence the test is passed.

b) Empty Password:- If user doesn't provide any appropriate password in password field, we will receive a pop-up message as please enter password and we expected the same result, hence the test is passed.

Order Product's

a) Empty Address Details: - If user doesn't provide any valid address detail's in appropriate field's, we will receive a pop-up message as please enter address detail's, and we expected the same result, hence the test is passed.

4.1.6 Inference:

Admin Login

In our project, the admin login is crucial because they oversee the entire project and control all user-related matters. Additionally, administration is crucial to our project.

Manage Categories

To make it easier for users to purchase things, admin should include the appropriate categories. Products will be added by the admin to the appropriate categories.

Manage Products

The administrator is in charge of managing the products in their relevant categories and can view the entire product inventory.

View Orders

View orders in depends on users adding orders as needed and requesting products. and the admin oversees its management.

User Registration

Before logging in, users must first register by providing their contact information, including their username, password, phone number, address, and email.

User Login

As he receives a username and password above for simple access to place orders for products, the user can log in using these data.

Order Products

The user can place orders for all the products after viewing them using the accurate information.

View Products

As soon as a user logs in using the proper information, they can browse the products and their categories before purchasing.

Offers

The user receives a coupon code as soon as they place an order for products.

Introduction

The aim of Online Medical Store is to automate the existing manual system with the help of computerized equipment and full-fledged computer software and meet the requirements of the customers so that their valuable data/information can be stored for a longer period of time with easy access and manipulation of the same. The required software and hardware is readily available and easy to use. An online medicine warehouse as described above can lead to an error-free, secure, reliable and fast management system.

In this project, users can view the medicines online and place orders at any time. The administrator can add the products and view the order list. The organization can keep computerized records without redundant entries. This means that people are not distracted by information that is not relevant while they are able to access the information. Basically, the project describes how to ensure good performance and better services for customers.

Proposed System: -

The main advantage of the proposed the system is the user can easily find out the availability of particular medicine through own devices. This system also shows the nearest pharmaceutical shops on their devices. This system can be divided into various sections.

In this project user can view the medicines online and give the orders any time. Admin can add the products and view the order list.

CHAPTER-5

5.1BUSSINESS ASPECTS

5.1.1 Briefly describe the market and economic outlook of the capstone project for the industry

The purpose of Online Medical Store 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. The required software and hardware are easily available and easy to work with. Online Medical Store, as described above, can lead to error free, secure, reliable and fast management system.

In this project user can view the medicines online and give the orders any time. Admin can add the products and view the order list. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information: that is not relevant, while being able to reach the information Basically, the project describes how to manage for good performance and better services for the clients.

The purpose of Online Medical Store 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. The required software and hardware are easily available and easy to work with. Online Medical Store, as described above, can lead to error free, secure, reliable and fast management system.

5.1.2 Features

User can order medicine anytime.
Information is private and confidential
Convenient
Easy to use, update and maintain
Time-saving

5.1.3 How does the product/services fit into the competitive landscape?

Online medicine shopping can be very helpful for individuals who have limited mobility, live in remote areas, or have busy schedules. It allows them to easily purchase their medications from the comfort of their own homes and have them delivered directly to their doorstep. Online medicine shopping can be very helpful to users as it provides convenience, accessibility, and sometimes even cost savings. Users can order their medications from the comfort of their own homes and have them delivered directly to their doorstep. This is especially beneficial for those who have mobility issues or live in remote areas where access to pharmacies may be limited.

5.1.4 Possible Capstone projects clients/customers

a) Individual Consumers: People who require regular medications, over-the-counter drugs, or health and wellness products can be primary clients. Online platforms provide convenience, easy accessibility, and the option to order medications from the comfort of their homes.

b) Elderly or Disabled Individuals: Elderly people or those with limited mobility may find it challenging to visit physical pharmacies. Online medicine shopping platforms can cater to their needs by delivering medications directly to their doorstep.

c) Patients with Chronic Illnesses: Individuals with chronic conditions, such as diabetes, hypertension, asthma, or autoimmune disorders, often require long-term medications. Online medicine shopping offers them the convenience of getting regular refills without the hassle of repeated visits to a physical pharmacy.

d) Caregivers: Family members or caregivers responsible for managing the healthcare needs of their loved ones can benefit from online medicine shopping. It allows them to order medications and healthcare products on behalf of the patients they are caring for.

e) Busy Professionals: Professionals with demanding schedules may find it challenging to take time off work to visit a pharmacy. Online medicine shopping platforms enable them to conveniently order medications and have them delivered to their homes or workplaces.

5.2 FINANCIAL CONSIDERATIONS

5.2.1 Capstone project budget:₹44,189/-

Estimated Cost:	₹44,18
Labour Cost:	₹24,500
Material Cost:	₹8,444
Profit of the Project: ₹11,245	

Table:5.2.1.1

5.3 CONCLUSION AND RECOMMENDATIONS

5.3.1 Describe state of complement of capstone project

As per the Synopsis we will implement E-commerce online medicine shopping with all the modules like User module and Admin module project error free and user friendly.

5.3.2 Future Work

Develop the mobile Application.
Online Payment.
Order tracking process.

PROGRAM CODE

Index

```
<!doctype html>

<!--[if IE 9 ]><html class="ie9" lang="en"><![endif]-->

<!--[if (gt IE 9)|!(IE)]><!--><html lang="en"><!--<![endif]-->

<head>

<title>Online Medicine Shopping</title>

<meta charset="utf-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no"/>

 <!--stylesheet include-->

<link rel="stylesheet" type="text/css" media="all" href="css/bootstrap.min.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/camera.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/owl.carousel.css"/>

<link rel="stylesheet" type="text/css" media="all"
href="css/owl.transitions.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/jquery.custom-
scrollbar.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/style.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/custom.css"/>

<!--font include-->

<link href="css/font-awesome.min.css" rel="stylesheet"/>

<script src="js/modernizr.js"></script>
```

```
<?php
require_once "medicinehelper.php";
$helper = new MedicineHelper();
$msg    = '';
?>
</head>
<body>
<!--boxed layout-->
<div class="boxed_layout relative w_xs_auto">
<?php
require_once "header.php";
 ?>
<!--slider-->
<div class="camera_wrap m_bottom_0">
<div data-src="images/s1.png" data-custom-thumb="images/s1.png">
</div>
<div data-src="images/s2.png" data-custom-thumb="images/s2.png">
</div>
</div>
<!--content-->
<div class="page_content_offset">
<div class="container">
```

```
<h2 class="tt_uppercase m_bottom_20 color_dark heading1 animate_ftr">All
Products</h2>

<!--products-->

<section class="products_container clearfix m_bottom_25 m_sm_bottom_15">

<!--product item-->

<?php

echo $helper->getLatestProducts();

?>

<!--product item-->

</section>

</div>

<!--banners-->

<div class="row clearfix">

<div class="col-lg-4 col-md-4 col-sm-4"></div>

<div class="col-lg-4 col-md-4 col-sm-4"></div>

<div class="col-lg-4 col-md-4 col-sm-4"></div>

</div>

</div>

<!--markup footer-->

<?php

require_once "footer.php";

?>

</div>
```

```
<button class="t_align_c r_corners tr_all_hover animate_ftl" id="go_to_top"><i class="fa fa-angle-up"></i></button>

<!--scripts include-->

<script src="js/jquery-2.1.0.min.js"></script>

<script src="js/jquery-migrate-1.2.1.min.js"></script>

<script src="js/retina.js"></script>

<script src="js/camera.min.js"></script>

<script src="js/jquery.easing.1.3.js"></script>

<script src="js/waypoints.min.js"></script>

<script src="js/jquery.isotope.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/jquery.tweet.min.js"></script>

 <script src="js/jquery.custom-scrollbar.js"></script>

<script src="js/scripts.js"></script>

</body>

</html>
```

Admin login

```
<!doctype html>

<!--[if IE 9 ]><html class="ie9" lang="en"><![endif]-->

<!--[if (gt IE 9)|!(IE)]><!--><html lang="en"><!--<![endif]-->

<head>

<title>Home</title>

<meta charset="utf-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no"/>

<!--stylesheet include-->

<link rel="stylesheet" type="text/css" media="all"
href="../css/bootstrap.min.css"/>

<link rel="stylesheet" type="text/css" media="all" href="../css/camera.css"/>

<link rel="stylesheet" type="text/css" media="all"
href="../css/owl.carousel.css"/>

<link rel="stylesheet" type="text/css" media="all"
href="../css/owl.transitions.css"/>

<link rel="stylesheet" type="text/css" media="all" href="../css/jquery.custom-
scrollbar.css"/>

<link rel="stylesheet" type="text/css" media="all" href="../css/style.css"/>

<!--font include-->

<link href="../css/font-awesome.min.css" rel="stylesheet"/>

<script src="../js/modernizr.js"></script>

<script type="text/javascript">
```

```
function validate_form(){
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;
if(username==''){
alert("Please Enter User Name.");
return false;
}
else if(password==''){
alert("Please Enter Password.");
 return false;
}
}
</script>
</head>
<?php
require_once "adminhelper.php";
$helper = new AdminHelper();
$msg    = '';
if($_POST){
$msg = $helper->checkUser();
}
?>
<body>
```

```
<div class="boxed_layout relative w_xs_auto">
  <style>#footer{padding:2px 0 0;}</style>
<header role="banner">
<section class="h_top_part">
<div class="container">
<div class="row clearfix">
</div>
</div>
</section>
<section class="h_bot_part container">
<div class="clearfix row">
<div class="col-lg-5 t_md_align_c m_md_bottom_15">
<a class="logo d_md_inline_b" href="index.php">
<h1 style="font-size: 26px;padding: 20px;">Online Medicine Shopping</h1>
</a>
</div>
<div class="col-lg-7 clearfix t_sm_align_c">
<div class="clearfix t_sm_align_l f_left f_sm_none relative s_form_wrap
m_sm_bottom_15 p_xs_hr_0 m_xs_bottom_5">
<!--button for responsive menu-->
<button class="r_corners centered_db d_none tr_all_hover d_xs_block
m_xs_bottom_5 color_blue" id="menu_button">
<span class="centered_db r_corners"></span>
```

```
<span class="centered_db r_corners"></span>

<span class="centered_db r_corners"></span>

</button>

<!--main menu-->

<nav class="f_left f_xs_none d_xs_none m_md_right_30 m_sm_right_0"
role="navigation">

<ul class="horizontal_list main_menu type_2 clearfix">

 <li class="relative f_xs_none m_xs_bottom_5 m_left_10 m_xs_left_0"><a
class="tr_delay_hover color_dark tt_uppercase r_corners"
href="../index.php"><b>Back to Portal</b></a></li>

</ul>

</nav>

</div>

</div>

</div>

</section>

</header>

<!--content-->

<hr />

<div class="page_content_offset">

<div class="container">

<div class="row clearfix">

<!--left content column-->

<section class="col-lg-9 col-md-9 col-sm-9">
```

```
<div class="row clearfix">

<div class="col-lg-8 col-md-8 col-sm-8 m_xs_bottom_30">

<h2 class="tt_uppercase color_dark m_bottom_25">Admin Login</h2>

<?php

if($msg!='')

{

?>

<div class="alert_box r_corners error m_bottom_10">

<i class="fa fa-exclamation-triangle"></i><p><?php echo $msg; ?> </p>

</div>

<?php

}

?>

<form id="" action="" style="min-height: 380px;" method="post"
onSubmit="return validate_form();">

<ul>

<li class="clearfix m_bottom_15">

<div class="f_left half_column">

  <label for="cf_name" class="required d_inline_b m_bottom_5">User
Name</label>

  <input type="text" id="username" name="username" class="full_width
r_corners">

</div>

</li>
```

```
<li class="clearfix m_bottom_15">

<div class="f_left half_column">

<label for="cf_password" class="required d_inline_b
m_bottom_5">Password</label>

<input type="password" id="password" name="password" class="full_width
r_corners">

 </div>

</li>

<li>

<button class="button_type_4 bg_light_color_2 r_corners mw_0 tr_all_hover
color_dark">Login</button>

</li>

</ul>

</form>

</div>

</div>

</section>

</div>

</div>

</div>

<!--markup footer-->

<?php

require_once "footer.php";

?>
```

```
</section>

</div>

<button class="t_align_c r_corners tr_all_hover type_2 animate_ftl"
id="go_to_top"><i class="fa fa-angle-up"></i></button>

<!--scripts include-->

<script src="../js/jquery-2.1.0.min.js"></script>

<script src="../js/jquery-ui.min.js"></script>

<script src="../js/retina.js"></script>

<script src="../js/waypoints.min.js"></script>

<script src="../js/jquery.isotope.min.js"></script>

<script src="../js/owl.carousel.min.js"></script>

<script src="../js/jquery.custom-scrollbar.js"></script>

  <script src="../js/scripts.js"></script>

  </body>

</html>
```

User registration

```
<!doctype html>

<!--[if IE 9 ]><html class="ie9" lang="en"><![endif]-->

<!--[if (gt IE 9)|!(IE)]><!--><html lang="en"><!--<![endif]-->

<head>

<title>Online Medicine Shopping</title>

<meta charset="utf-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no"/>

<!--stylesheet include-->

<link rel="stylesheet" type="text/css" media="all" href="css/bootstrap.min.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/camera.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/owl.carousel.css"/>

<link rel="stylesheet" type="text/css" media="all"
href="css/owl.transitions.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/jquery.custom-
scrollbar.css"/>

<link rel="stylesheet" type="text/css" media="all" href="css/style.css"/>

<!--font include-->

<link href="css/font-awesome.min.css" rel="stylesheet"/>

<script src="js/modernizr.js"></script>

<script type="text/javascript">

function validate_form()

{
```

```
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;
var name = document.getElementById("name").value;
var email = document.getElementById("email").value;
var mobile = document.getElementById("mobile").value;
var validchar = /^[A-Z a-z]+$/;
if(username=='')
{
alert("Please Enter User Name.");
return false;
}
else if(password=='')
{
alert("Please Enter Password.");
return false;
}
else if(name=='')
{
alert("Please Enter Name.");
return false;
}
else if(!validchar.test(name))
{
```

```
 alert("Name should not be numeric.");
return false;
}
else if(email==")
{
alert("Please Enter Email Address.");
return false;
}
else if(validateEmail(email))
{
alert("Please Enter Valid Email Address.");
return false;
}
else if(mobile==")
{
alert("Please Enter Mobile Number.");
return false;
}
else if(isNaN(mobile))
{
alert("Mobile Number should be numeric.");
return false;
}
```

```
else if(checkInternationalPhone(mobile)==false)

{

alert("Please Enter a Valid Mobile Number");

return false;

}

}

function validateEmail(email)

{

var atpos  = email.indexOf("@");   // The indexOf() method returns the position of the first occurrence of a specified value in a string. // Default value of start is 0

//alert(atpos);

var dotpos = email.lastIndexOf(".");  // The lastIndexOf() method returns the position of the last occurrence of a specified value in a string. // Default value of start is 0

//alert(dotpos);

var atpos  = email.indexOf("gmail");   // The indexOf() method returns the position of the first occurrence of a specified value in a string. // Default value of start is 0



//alert(atpos);

if((atpos<1) || (dotpos<(atpos+2)) || (dotpos+2>=email.length))

{
```

```
return true;
}
else
{
return false;
}
}
// Declaring required variables
var digits = "0123456789";
// non-digit characters which are allowed in phone numbers
var phoneNumberDelimiters = "()- ";
// characters which are allowed in international phone numbers
// (a leading + is OK)
var validWorldPhoneChars = phoneNumberDelimiters + "+";
// Minimum no of digits in an international phone no.
var minDigitsInIPhoneNumber = 10;
function isInteger(s)
{   var i;
```

```
for (i = 0; i < s.length; i++)

{

// Check that current character is number.

var c = s.charAt(i);

if (((c < "0") || (c > "9"))) return false;

}

// All characters are numbers.

return true;

}

function trim(s)

{   var i;

var returnString = "";

// Search through string's characters one by one.

 // If character is not a whitespace, append to returnString.

for (i = 0; i < s.length; i++)

{

// Check that current character isn't whitespace.

var c = s.charAt(i);

if (c != " ") returnString += c;

}

return returnString;

}
```

```
function stripCharsInBag(s, bag)
{   var i;
var returnString = "";
// Search through string's characters one by one.
// If character is not in bag, append to returnString.
for (i = 0; i < s.length; i++)
{
// Check that current character isn't whitespace.
var c = s.charAt(i);
if (bag.indexOf(c) == -1) returnString += c;
}
return returnString;
}
function checkInternationalPhone(strPhone){
var bracket=3;
strPhone=trim(strPhone);
if(strPhone.indexOf("+")>1) return false;
if(strPhone.indexOf("-")!=-1)bracket=bracket+1;
if(strPhone.indexOf("(")!=-1 && strPhone.indexOf("(")>bracket)return false;
var brchr=strPhone.indexOf("(");
if(strPhone.indexOf("(")!=-1 && strPhone.charAt(brchr+2)!=")")return false;
if(strPhone.indexOf("(")==-1 && strPhone.indexOf(")")!=-1)return false;
s=stripCharsInBag(strPhone,validWorldPhoneChars);
```

```
return (isInteger(s) && s.length >= minDigitsInIPhoneNumber);

}

</script>

</head>

<?php

require_once "medicinehelper.php";

$helper = new MedicineHelper();

$msg    = '';

if($_POST)

{

$msg = $helper->regUsers();

}

?>

    <body>

<div class="boxed_layout relative w_xs_auto">

<?php

require_once "header.php";

?>

<!--breadcrumbs-->


<section class="breadcrumbs">


<div class="container">
```

```
<ul class="horizontal_list clearfix bc_list f_size_medium">
<li class="m_right_10 current"><a href="#" class="default_t_color">Home<i class="fa fa-angle-right d_inline_middle m_left_10"></i></a></li>
<li><a href="#" class="default_t_color">Register</a></li>
</ul>
</div>
</section>
<!--content-->
<div class="page_content_offset">
<div class="container">
<div class="row clearfix">
<!--left content column-->
<section class="col-lg-9 col-md-9 col-sm-9">
<h2 class="tt_uppercase color_dark m_bottom_25">Register</h2>
<div class="row clearfix">
<div class="col-lg-8 col-md-8 col-sm-8 m_xs_bottom_30">
<?php
if($msg!='')
{
?>
<div class="alert_box r_corners color_green success m_bottom_10">
<i class="fa fa-smile-o"></i><p><?php echo $msg; ?> </p>
```

```
</div>
<?php
}


?>
<form method="post" action="" onSubmit="return validate_form();">
<ul>
<li class="clearfix m_bottom_15">
<div class="f_left half_column">
<label for="cf_name" class="required d_inline_b m_bottom_5">User Name</label>
<input type="text" id="username" name="username" class="full_width r_corners">
</div>
<div class="f_left half_column">
<label class="required d_inline_b m_bottom_5">Password</label>
<input type="password" id="password" name="password" class="full_width r_corners">
</div>
</li>
<li class="clearfix m_bottom_15">
<div class="f_left half_column">
<label for="cf_name" class="required d_inline_b m_bottom_5">Name</label>
<input type="text" id="name" name="name" class="full_width r_corners">
```

```
</div>

<div class="f_left half_column">

<label for="cf_email" class="required d_inline_b m_bottom_5">Email</label>

                                                                    <input
type="text" id="email" name="email" class="full_width r_corners">

</div>

</li>

<li class="m_bottom_15">

<div class="f_left half_column">

<label for="cf_name" class="required d_inline_b m_bottom_5">Mobile</label>

<input type="text" id="mobile" name="mobile" class="full_width r_corners"
size="10">

</div>

</li>

<div class="clearfix"></div>

<li class="clearfix m_bottom_15">

<input type="submit" class="button_type_12 r_corners bg_scheme_color
color_light tr_delay_hover d_inline_b f_size_large" style="margin-top: 8px;"
value="Register"/>

</li>

</ul>

</form>

</div>
```

```
</div>

</section>

</div>

</div>

</div>

<!--markup footer-->

<?php

require_once "footer.php";

?>

</div>

<button class="t_align_c r_corners tr_all_hover type_2 animate_ftl"
id="go_to_top"><i class="fa fa-angle-up"></i></button>

<!--scripts include-->

<script src="js/jquery-2.1.0.min.js"></script>

<script src="js/jquery-ui.min.js"></script>

<script src="js/retina.js"></script>

<script src="js/waypoints.min.js"></script>

<script src="js/jquery.isotope.min.js"></script>

<script src="js/jquery.tweet.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/jquery.custom-scrollbar.js"></script>

<script src="js/styleswitcher.js"></script>

<script src="js/colorpicker.js"></script>
```

<script src="js/scripts.js"></script>

</body>

</html>

References

[1] http://en.wikipedia.org/wiki/PHP for Php.

[2] http://www.hotscripts.com/category/php/ for Php

[3] http://www.mysql.com/click.php?e=35050 for MySql.

[4] http://www.w3schools.com for information on HTML.

Books:-

[1] “Web Programming”, by ‘Chris Bates’ Wiley Dreamtech India, 2nd Edition.

[2] “Software Engineering”, Ian Somerville, Sixth Edition, Pearson Education Ltd.

[3] “HTML Complete References” Easy steps to develop web pages.

[4] “PHP Complete Reference”

Appendices

Chapter 3

Chapter 4

Chapter 5

Figure number	Figure name	Page number
Fig 2.1.1.1	System architecture	6
Fig 2.3.1.1	Chosen system design	16
Fig 2.3.2.1	Data flow diagram level 0	17
Fig 2.3.2.2	Data flow diagram level 1 Admin	17