**CHAPTER 1**

**INTRODUCTION**

Today the world’s most forward-looking medical agency are trying to provide more reliable and accurate services in their field, offering services to the customers and employees with all the available choices in their interest. Every Shop nowadays is trying to computerize its activities to provide better services to its customers. The aim is to automate its existing manual system by the help of computerized equipments 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.

This project, "ONLINE MEDICAL STORE MANAGEMENT SYSTEM" also a step towards offering more or less the similar features. This system enables to manage and record the activities of whole medical Shop of multi-facility skills only. This system organize their daily activities like billing, tablets information, stock details and more. In present trend this application is used in every medical shops. This system will save time and increase work efficiency.

Medical Shop Management System is an application project developed for medical shops. This system is a field concerned with purchasing and selling medicines, maintaining their inventory, generating sales invoices about medicines. It requires more time and effort when all procedures are performed manually. Thus, in order to reduce time consumption and human effort the Medical Shop Management System application can be applied in medicals where manual procedure exists. The purpose of this project this is to reduce time consumption and human effort. This application provides user friendly interface as well.

**Problem Statement**:

First selecting the suitable medicine for the type of illness is usually take time and makes the patient or customer waiting. Therefore, the time is waste for the customer to be waiting. Second. for the medicines stock management. The pharmacist must check it.

**CHAPTER 2**

**REQUIREMENTS ANALYSIS**

The requirement analysis specifies the requirements needed to develop a graphic project. In this phase, we collect the requirements needed for designing the project. The requirements collected are then analysed and carried to the next phase.

**2.1 SOFTWARE REQUIREMENTS:**

* Operating System: Windows 7 or above
* Script Language: PHP and MYSQL
* Front-end Development: HTML,CSS
* Back-end Development: PHP,MYSQL

**2.2 HARDWARE REQUIREMENTS**

* Processor – Pentium IV or above
* RAM – 2 GB or more
* Hard disk – 3 GB or more

**CHAPTER 3**

**DESIGN**

**3.1 ER DIAGRAM**

An **Entity – Relationship model** (**ER model**) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

The E-R diagram of our “ONLINE MEDICAL STORE MANAGEMENT SYSTEM“ contains 6 Entities.

(items,users,orders,orderdetail,ticket,ticketdetails)

In the E-R Diagram USERS: is the entity where we can enter and the details about user id and name.

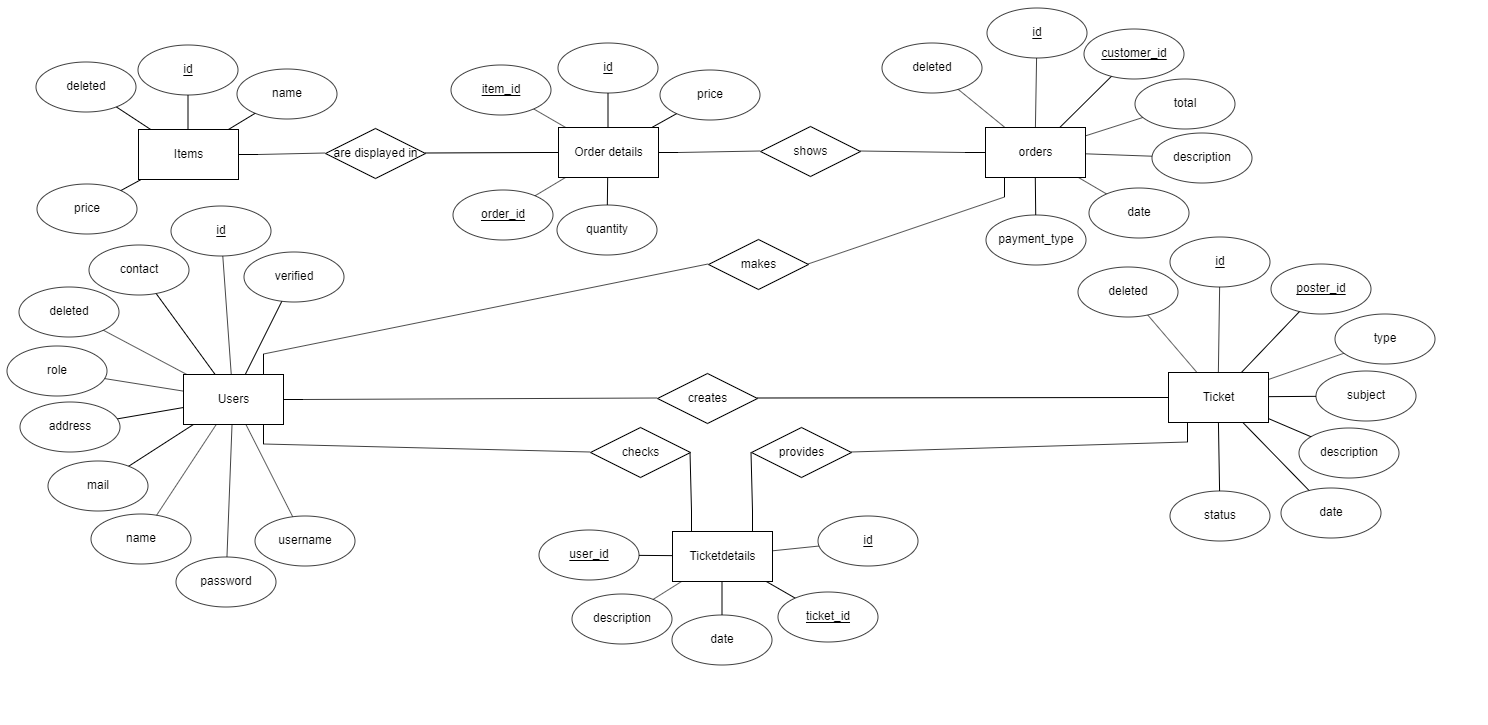
ITEMS: is the entity where the admin can add or delete the medicines.

ORDERS DETAILS: is the entity where users can see order details and order date and quantity and price .

ORDER:is the entity where the user can able to make the order of the medicines.

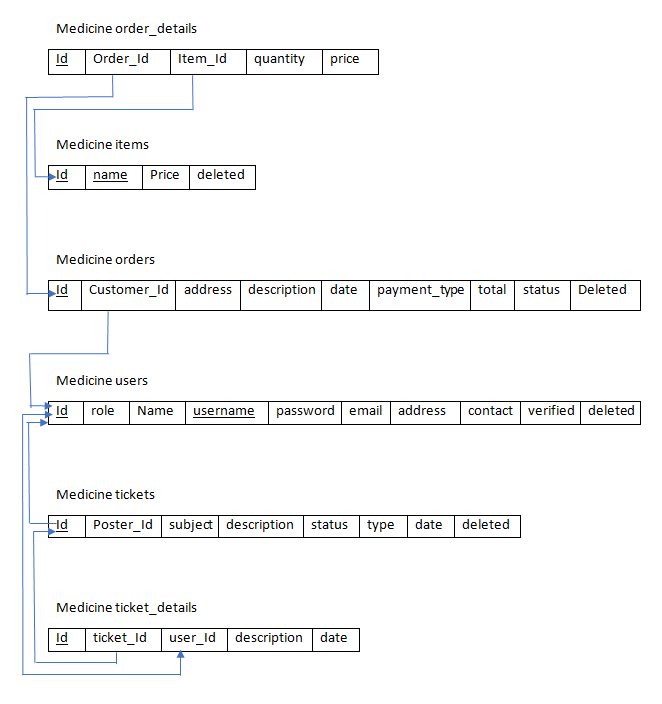
TICKETS: is the entity where the user can able to ask questions or giving feedback or by supporting the medicine.

TICKET DETAILS: is the entity to know the details like description and other details.



*Figure 3.1: Entity – Relational diagram of Medical Store*

**3.2 SCHEMA DIAGRAM**

****

*Figure 3.2: Relational Schema Diagram of Medical Store Management*

The term "**schema**" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of **relational** databases).

**3.3 USE CASE DIAGRAM**

ADMIN

*Figure 3.3: Use case diagram of medical store management*

The main purpose of a use case diagram is to show who interacts with your system, and the main goals they achieve with it.

**CHAPTER 4**

**IMPLEMENTATION**

**4.1 INTRODUCTION TO FRONT END TOOL**

**4.1.1 HTML & CSS**

**HTML: HTML** stands for Hyper Text Markup Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. Markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of markup (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text.  
HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991. The first ever version of HTML was HTML 1.0 but the first standard version was HTML 2.0 which was published in 1999.

**CSS**: **C**ascading **S**tyle **S**heets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of HTML that makes up each web page. CSS is easy to learn and understood but it provides powerful control over the presentation of an HTML document.

.

**4.2 INTRODUCTION TO BACK END TOOL**

**4.2.1 PHP & MYSQL**

**PHP**: The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server side scripting language designed specifically for web development. PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc. There are also many hash functions available in PHP to encrypt user’s data that makes PHP secure and reliable to be used as a server-side scripting language.

**MYSQL**: MySQL is a Relational DataBase Management System (RDBMS).

RDBMS means R--DB--MS.

- DB stands for Database, a repository for the information store.

1. The data in a database is organized into tables, and each table is organized into rows and columns.
2. Each row in a table is called a record. A record may contains several pieces (called fields) of information, and each column in a table is known as a field.

-MS stands for Management System, the software that allows you to insert, retrieve, modify, or delete records.

-R stands for Relational, indicates a particular kind of DBMS that is good at relating information stored in one table to information stored in another table by looking for elements common to each of them. Relational DBMS has the advantage of efficient storage, and retrieval mechanisms for data, and uses normalization process during design of RDBMS. Database normalization process is beyond the scope of this article, and several references are available.

MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. The server operating systems is usually a Linux (like Redhat 9.0 etc.) or Windows 2000 operating system. Typically mySQL is supported on Windows XP, Windows Server 2003, Red Hat Fedora Linux, and Debian Linux, and others. As with any other client/server application, MySQL is a multi-user database system, meaning several users can access the database simultaneously. Here:

-The server (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.

- Clients are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards based SQL (SQL stands for Structured Query Language) language. The client program may contact the server programmatically (meaning a program call the server during execution) or manually. For example, when you are issuing commands over a telnet session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

* 1. **CONNECTIVITY OF THE DATABASE**

**There are three ways of working with MySQl and PHP**

1. MySQLi (object-oriented)
2. MySQLi (procedural)
3. PDO

**Connecting to MySQL database using PHP**

1. **Using MySQLi object-oriented procedure**: We can use the MySQLi object-oriented procedure to establish a connection to MySQL database from a PHP script.

**Syntax**:

<?php

$servername = "localhost";

$username = "username";

$password = "password";

**// Creating connection**

$conn = new mysqli($servername, $username, $password);

**// Checking connection**

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

?>

**Output:**  
https://media.geeksforgeeks.org/wp-content/uploads/Screen-Shot-2017-11-20-at-2.23.19-PM.png

**Explanation**: We can create an instance of the mysqli class providing all the necessary details required to establish the connection such as host, username, password etc. If the instance is created successfully then the connection is successful otherwise there is some error in establishing connection.

**4.4 MODULES**

1. **Administrator login**

<?php

session\_start();

if(isset($\_SESSION['admin\_sid']) || isset($\_SESSION['customer\_sid']))

{

header("location:index.php");

}

else{

?>

</head>

<body class="cyan">

<!-- Start Page Loading -->

<div id="loader-wrapper">

<div id="loader"></div>

<div class="loader-section section-left"></div>

<div class="loader-section section-right"></div>

</div>

<!-- End Page Loading -->

<div id="login-page" class="row">

<div class="col s12 z-depth-4 card-panel">

<form method="post" action="routers/router.php" class="login-form" id="form">

<div class="row">

<div class="input-field col s12 center">

<p class="center login-form-text">Login to MedEasy</p>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-social-person-outline prefix"></i>

<input name="username" id="username" type="text">

<label for="username" class="center-align">Username</label>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-action-lock-outline prefix"></i>

<input name="password" id="password" type="password">

<label for="password">Password</label>

</div>

</div>

<div class="row">

<a href="javascript:void(0);" onclick="document.getElementById('form').submit();" class="btn waves-effect waves-light col s12">Login</a>

</div>

<div class="row">

<div class="input-field col s6 m6 l6">

<p class="margin medium-small"><a href="register.php">Register Now!</a></p>

</div>

</div>

</div>

</form>

</div>

</div>

<!-- jQuery Library -->

<script type="text/javascript" src="js/plugins/jquery-1.11.2.min.js"></script>

<!--materialize js-->

<script type="text/javascript" src="js/materialize.min.js"></script>

<!--scrollbar-->

<script type="text/javascript" src="js/plugins/perfect-scrollbar/perfect- scrollbar.min.js"></script>

<!--plugins.js - Some Specific JS codes for Plugin Settings-->

<script type="text/javascript" src="js/plugins.min.js"></script>

<!--custom-script.js - Add your own theme custom JS-->

<script type="text/javascript" src="js/custom-script.js"></script>

</body>

</html>

<?php

}

?>

1. **User Sign-Up Page**

<?php

session\_start();

if(isset($\_SESSION['admin\_sid']) || isset($\_SESSION['customer\_sid']))

{

header("location:index.php");

}

else{

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1.0, user-scalable=no">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="msapplication-tap-highlight" content="no">

<title>Register</title>

<!-- Favicons-->

<link rel="icon" href="images/favicon/favicon-32x32.png" sizes="32x32">

<!-- Favicons-->

<link rel="apple-touch-icon-precomposed" href="images/favicon/apple-touch-icon-152x152.png">

<!-- For iPhone -->

<meta name="msapplication-TileColor" content="#00bcd4">

<meta name="msapplication-TileImage" content="images/favicon/mstile-144x144.png">

<!-- For Windows Phone -->

<!-- CORE CSS-->

<link href="css/materialize.min.css" type="text/css" rel="stylesheet" media="screen,projection">

<link href="css/style.min.css" type="text/css" rel="stylesheet" media="screen,projection">

<!-- Custome CSS-->

<link href="css/custom/custom.min.css" type="text/css" rel="stylesheet" media="screen,projection">

<link href="css/layouts/page-center.css" type="text/css" rel="stylesheet" media="screen,projection">

<link href="js/plugins/perfect-scrollbar/perfect-scrollbar.css" type="text/css" rel="stylesheet" media="screen,projection">

<style type="text/css">

.input-field div.error{

position: relative;

top: -1rem;

left: 0rem;

font-size: 0.8rem;

color:#FF4081;

-webkit-transform: translateY(0%);

-ms-transform: translateY(0%);

-o-transform: translateY(0%);

transform: translateY(0%);

}

.input-field label.active{

width:100%;

}

.left-alert input[type=text] + label:after,

.left-alert input[type=password] + label:after,

.left-alert input[type=email] + label:after,

.left-alert input[type=url] + label:after,

.left-alert input[type=time] + label:after,

.left-alert input[type=date] + label:after,

.left-alert input[type=datetime-local] + label:after,

.left-alert input[type=tel] + label:after,

.left-alert input[type=number] + label:after,

.left-alert input[type=search] + label:after,

.left-alert textarea.materialize-textarea + label:after{

left:0px;

}

</style>

</head>

<body class="cyan">

<!-- Start Page Loading -->

<div id="loader-wrapper">

<div id="loader"></div>

<div class="loader-section section-left"></div>

<div class="loader-section section-right"></div>

</div>

<!-- End Page Loading -->

<div id="login-page" class="row">

<div class="col s12 z-depth-4 card-panel">

<form class="formValidate" id="formValidate" method="post" action="routers/register-router.php" novalidate="novalidate" class="col s12">

<div class="row">

<div class="input-field col s12 center">

<h4>Register</h4>

<p class="center">Join us now!</p>

<p class="center">We supply what you need!</p>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-social-person-outline prefix"></i>

<input name="username" id="username"type="text”data-error=".errorTxt1">

<label for="username" class="center-align">Username</label>

<div class="errorTxt1"></div>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-social-person prefix"></i>

<input name="name" id="name" type="text" data-error=".errorTxt2">

<label for="name" class="center-align">Name</label>

<div class="errorTxt2"></div>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-action-lock-outline prefix"></i>

<input name="password" id="password" type="password" data-error=".errorTxt3">

<label for="password">Password</label>

<div class="errorTxt3"></div>

</div>

</div>

<div class="row margin">

<div class="input-field col s12">

<i class="mdi-communication-phone prefix"></i>

<input name="phone" id="phone" type="text" data-error=".errorTxt4" minlength="9" maxlength="10">

<label for="phone">Phone</label>

<div class="errorTxt4"></div>

</div>

</div>

<div class="row">

<div class="input-field col s12">

<a href="javascript:void(0);" onclick="document.getElementById('formValidate').submit();" class="btn waves-effect waves-light col s12">Register</a>

</div>

<div class="input-field col s12">

<p class="margin center medium-small sign-up">Already have an account? <a href="login.php">Login</a></p>

</div>

</div>

</form>

</div>

</div>

<!-- jQuery Library -->

<script type="text/javascript" src="js/plugins/jquery-1.11.2.min.js"></script>

<!--materialize js-->

<script type="text/javascript" src="js/materialize.min.js"></script>

<!--scrollbar-->

<script type="text/javascript" src="js/plugins/perfect-scrollbar/perfect-scrollbar.min.js"></script>

<script type="text/javascript" src="js/plugins/jquery-validation/jquery.validate.min.js"></script>

<script type="text/javascript" src="js/plugins/jquery-validation/additional-methods.min.js"></script>

errorElement : 'div',

errorPlacement: function(error, element) {

var placement = $(element).data('error');

if (placement) {

$(placement).append(error)

} else {

error.insertAfter(element);

}

}

});

</script>

</body>

</html>

<?php

}

?>

**CHAPTER 5**

**TESTING**

**5.1 TESTING**

Testing is the process of executing a program to find the errors. A good test has the high probability of finding a yet undiscovered error. A test is vital to the success of the system. System test makes a logical assumption that if all parts of the system are correct, then goal will be successfully achieved.

**5.2 TYPES OF TESTING**

5.2.1 Module Testing.

5.2.2 Integration Testing.

**Module Testing**

Module testing is the testing of complete code objects as produced by the complier when built from source.

A library may be composed of a single complied object or several complied objects. There is only a slight difference between unit testing and module testing. Modules are fully formed chunks of coherent source code that can typically be tested by driving a few functions signatures with various stimuli. On the other hand, unit testing (which is considered as part of the implementation phase for this software development process) may involve testing one small part of a function that will never formally implement any function interface.

As a result of modules being more self-contained, module testing will likely require less testing infrastructure such as test harness and test stubs. The testing of modules could perhaps even be automated so that they can be included in regression test suites or a acceptance test suites.

**Integration Testing**

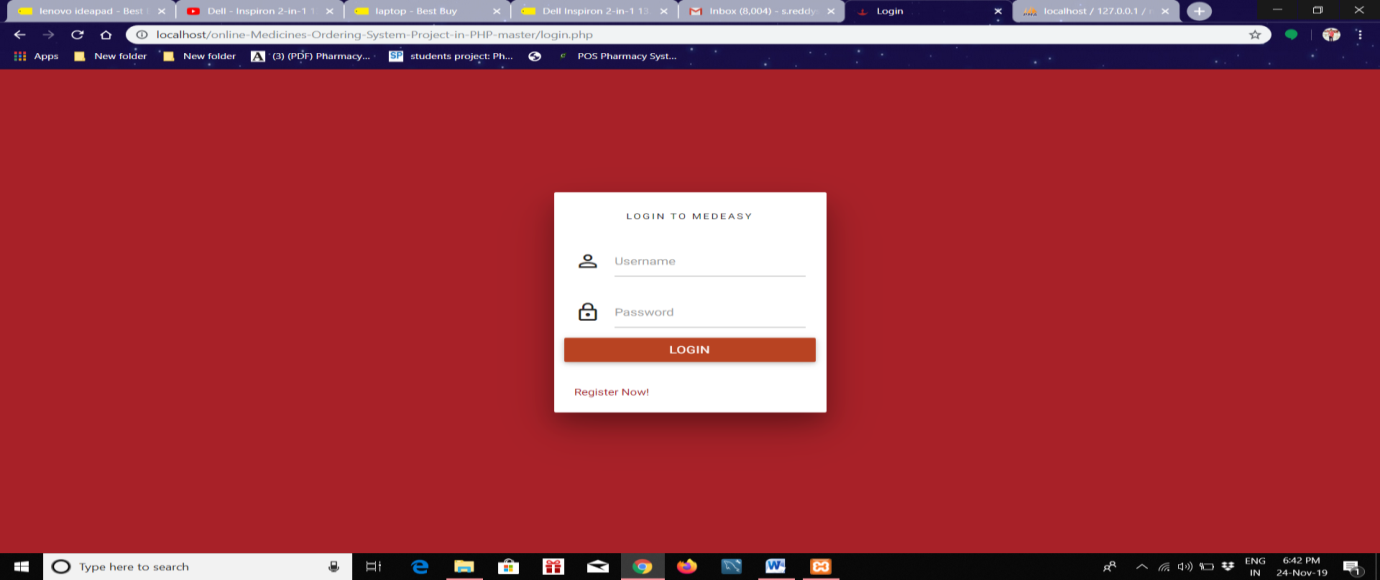
Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Input Data** | **Expected Output** | **Actual Output** | **Status** |
| 1 | Login Page | Username  Password | Login  successful | Login successful | Pass |
| 2 | Register | Username, Name, Password, Phone | Register  successful | Register  successful | Pass |
| 3 | Order Medicines | Name,  Item price, quantity | Order | Order | Pass |
| 4 | Order details | Select Payment,  Address, Confirm Order | Order confirmed | Order confirmed | Pass |
| 5 | Edit details | User name, Name, Email, Password, Contact, Address | Update Details | Update Details | Pass |

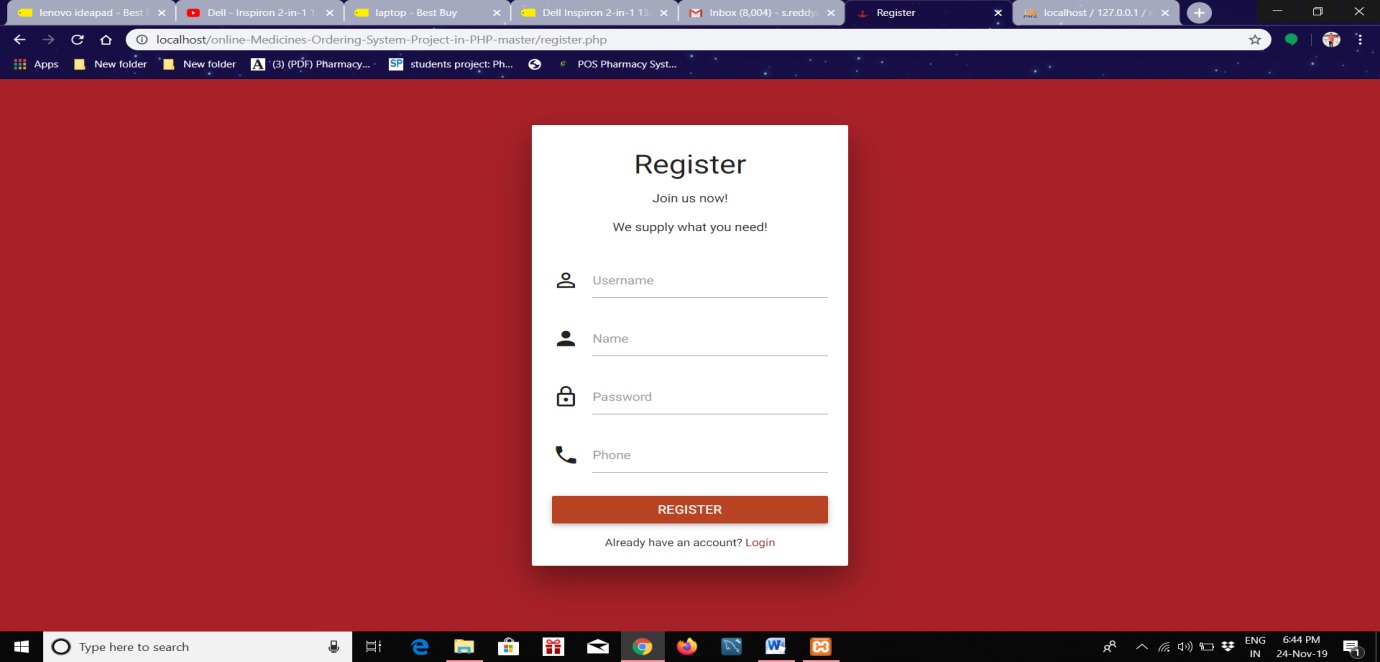
Figure 5.1: Integration testing table

**CHAPTER 6**

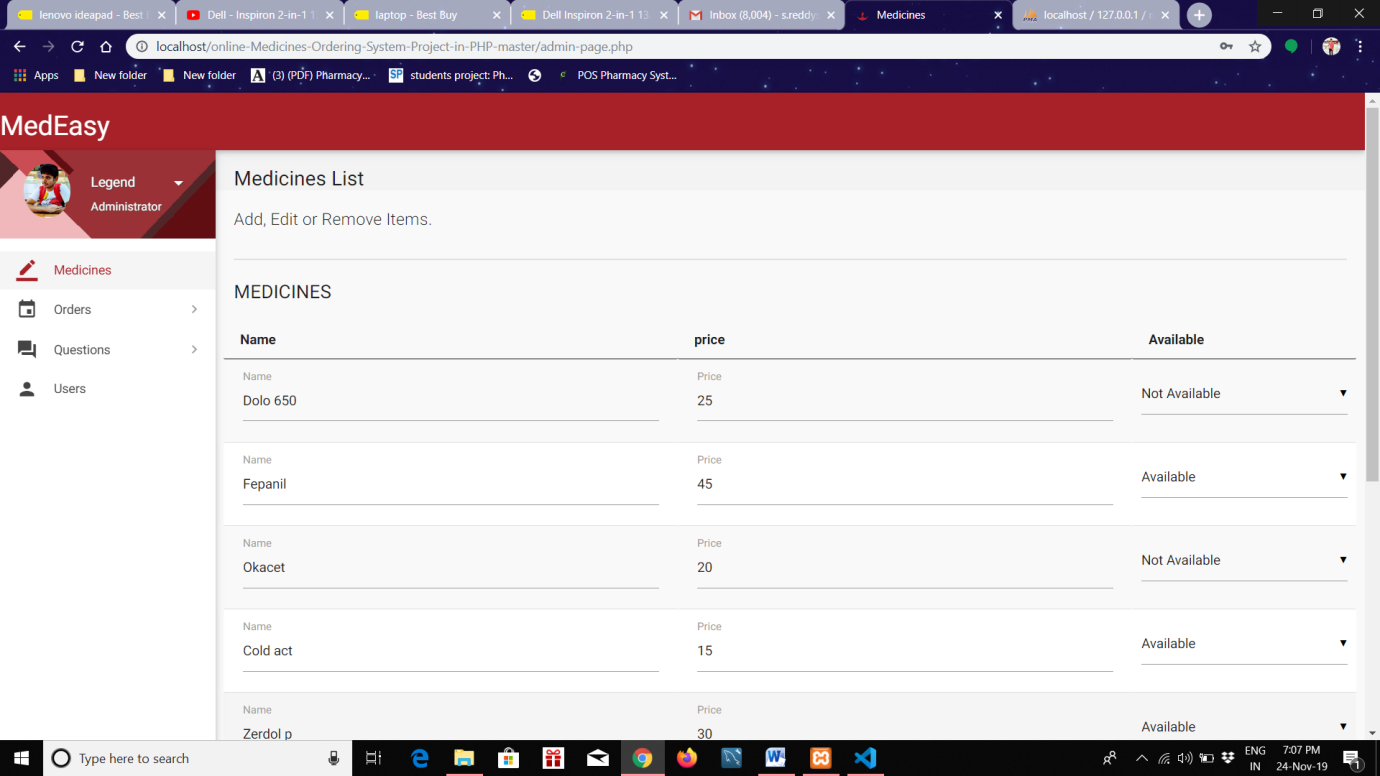
**SNAPSHOTS**



Snapshot 6.1: “Login Page”

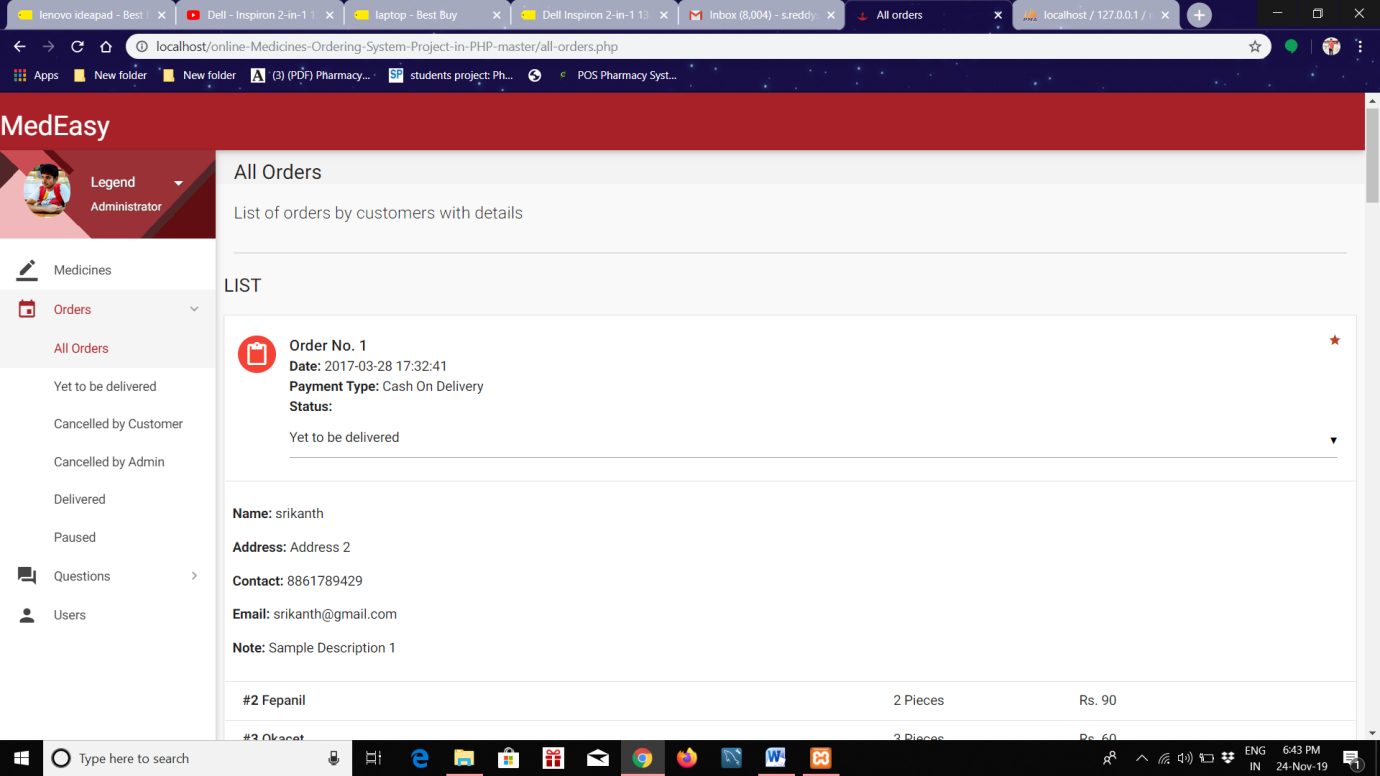
This Login Page is used to login for administration and customer. So that they can able to browse the application.

Snapshot 6.2: “Register form” In this register form a new customer can able to create a account or able to register for this application.



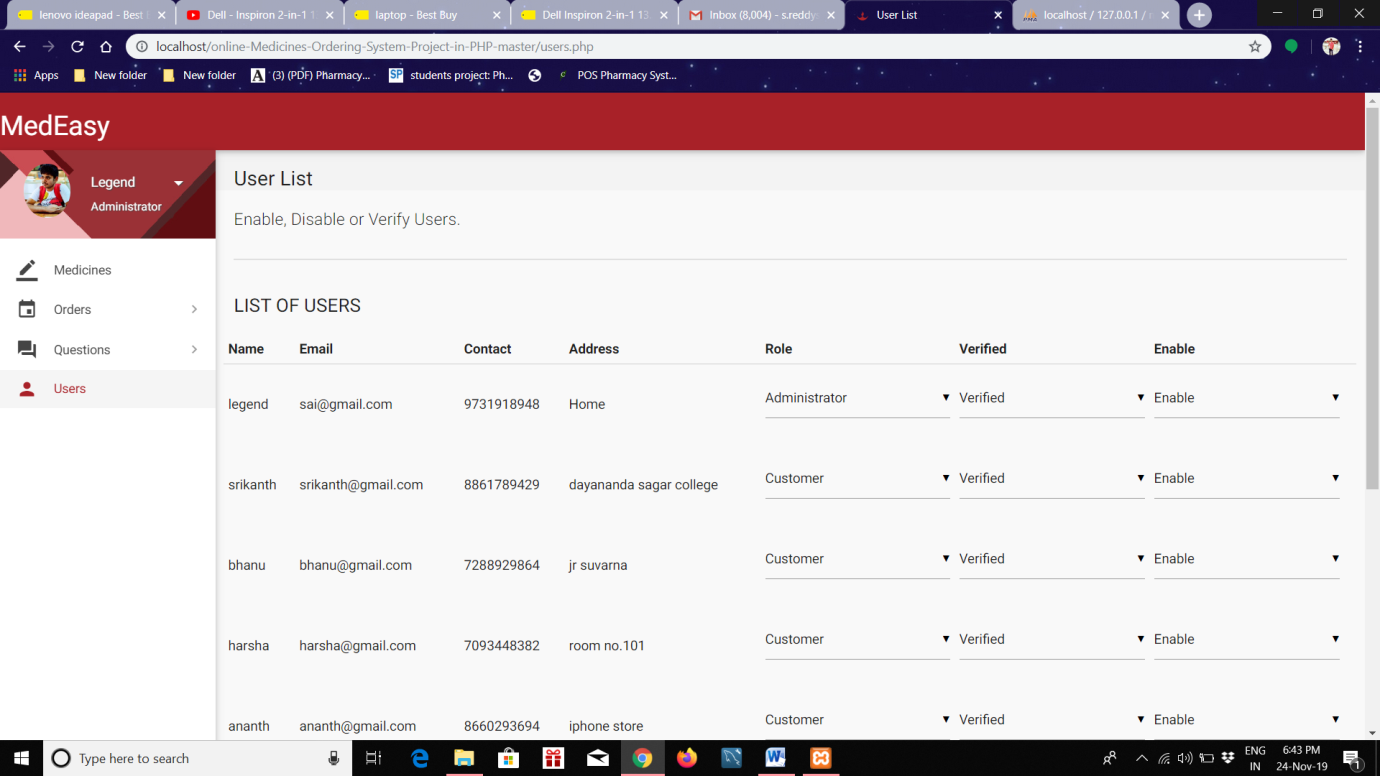
Snapshot 6.3: “Admin home page”

In this page admin can have many options like he can able to add medicines or delete medicines and also can able to update the details.



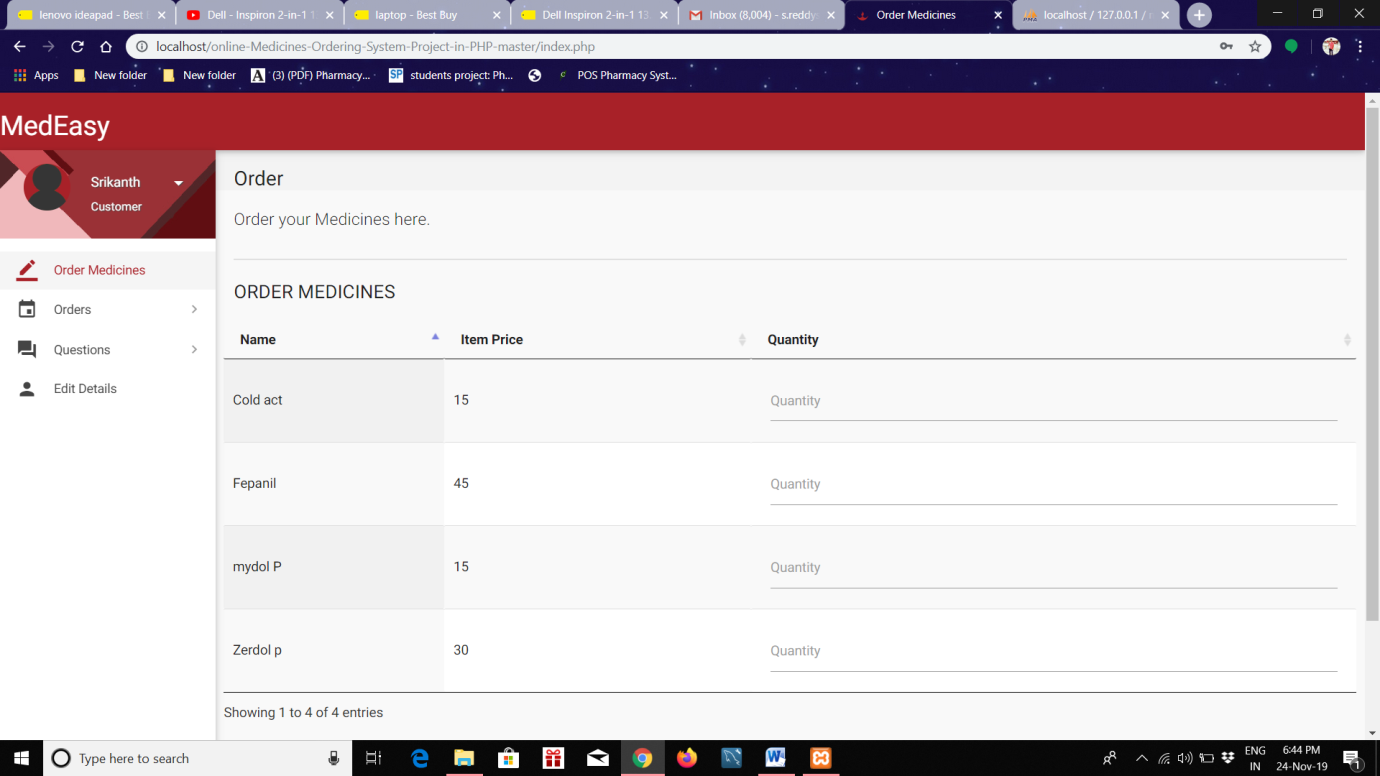
Snapshot 6.4: “Orders Received”

The orders page is used to check the total orders received to the admin.



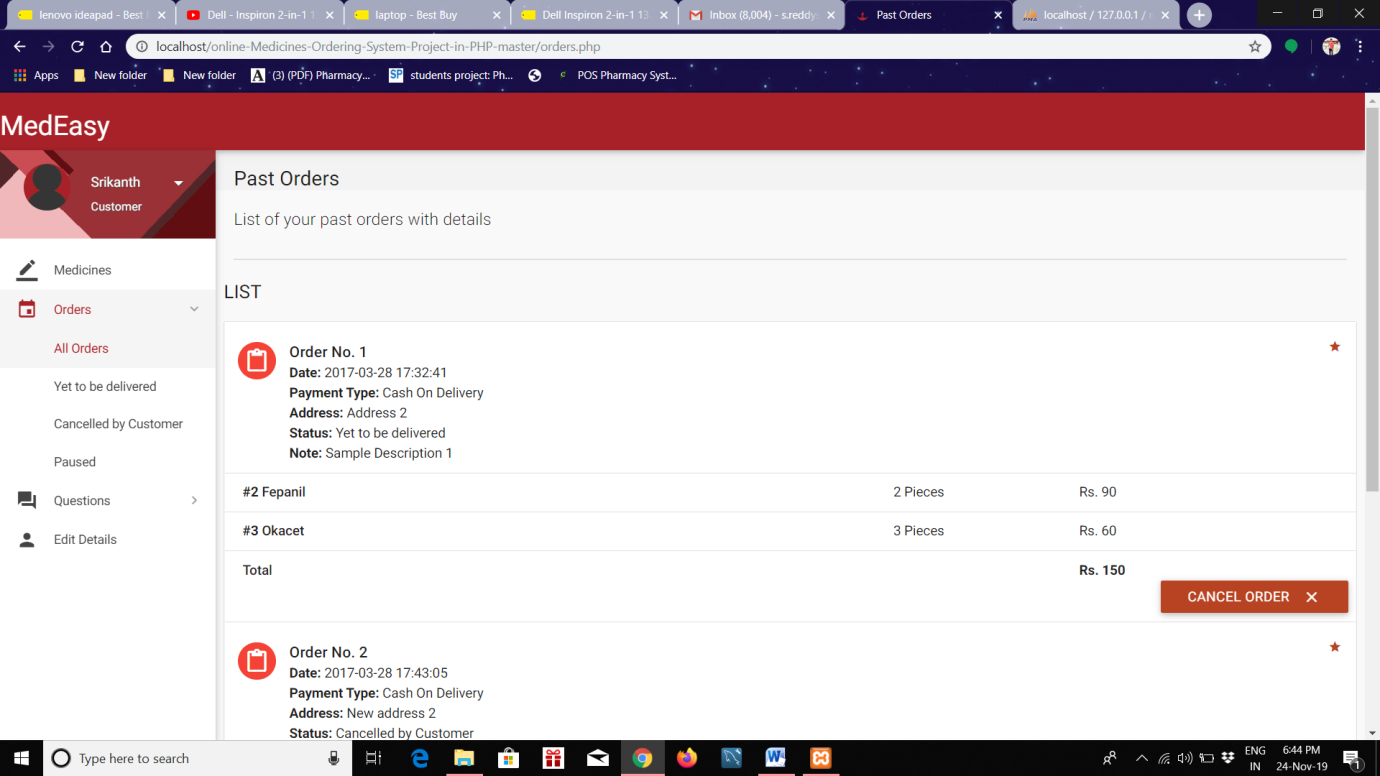
Snapshot 6.5: “Users Page”

This page is used to add and modify the details of the users.



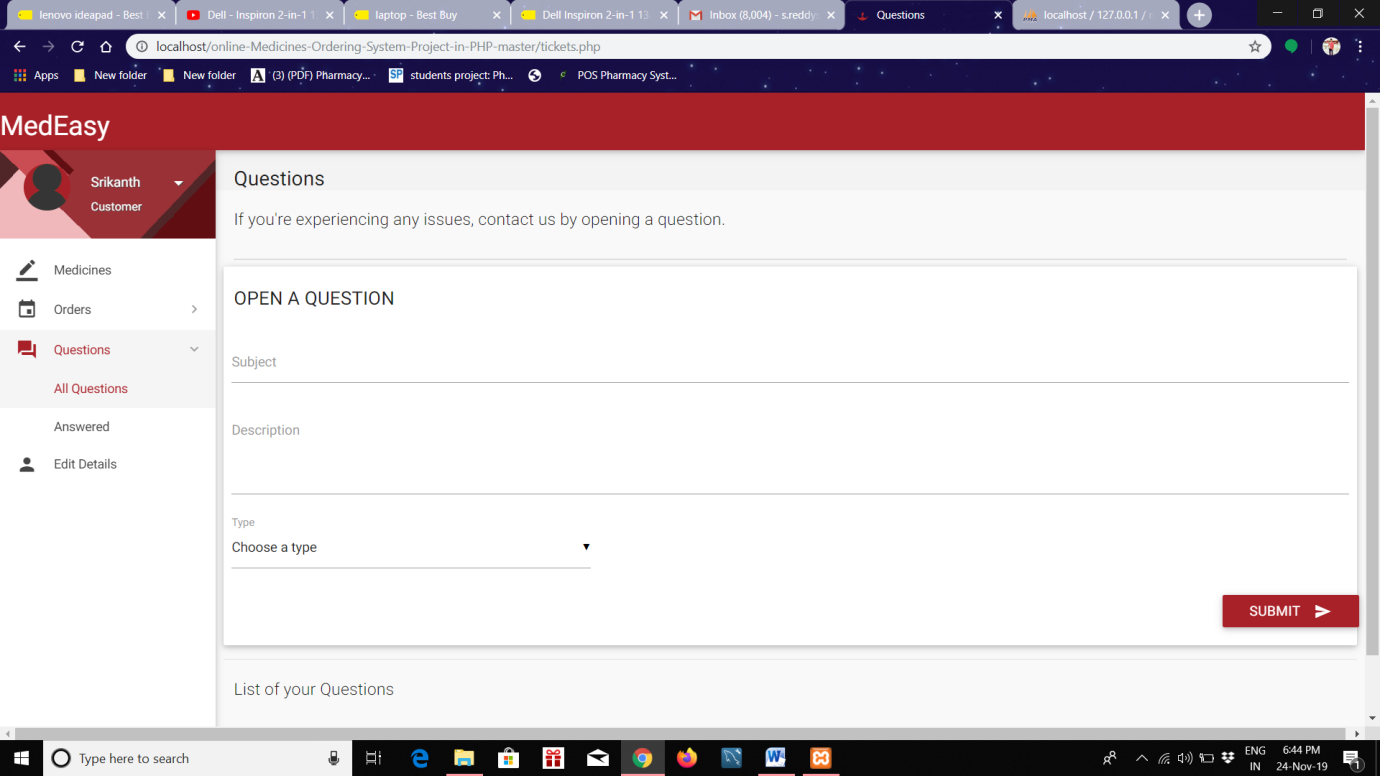
Snapshot 6.6: “Customer home page”

This is used by customer to order the medicines. They can add quantity number for the medicines they required.



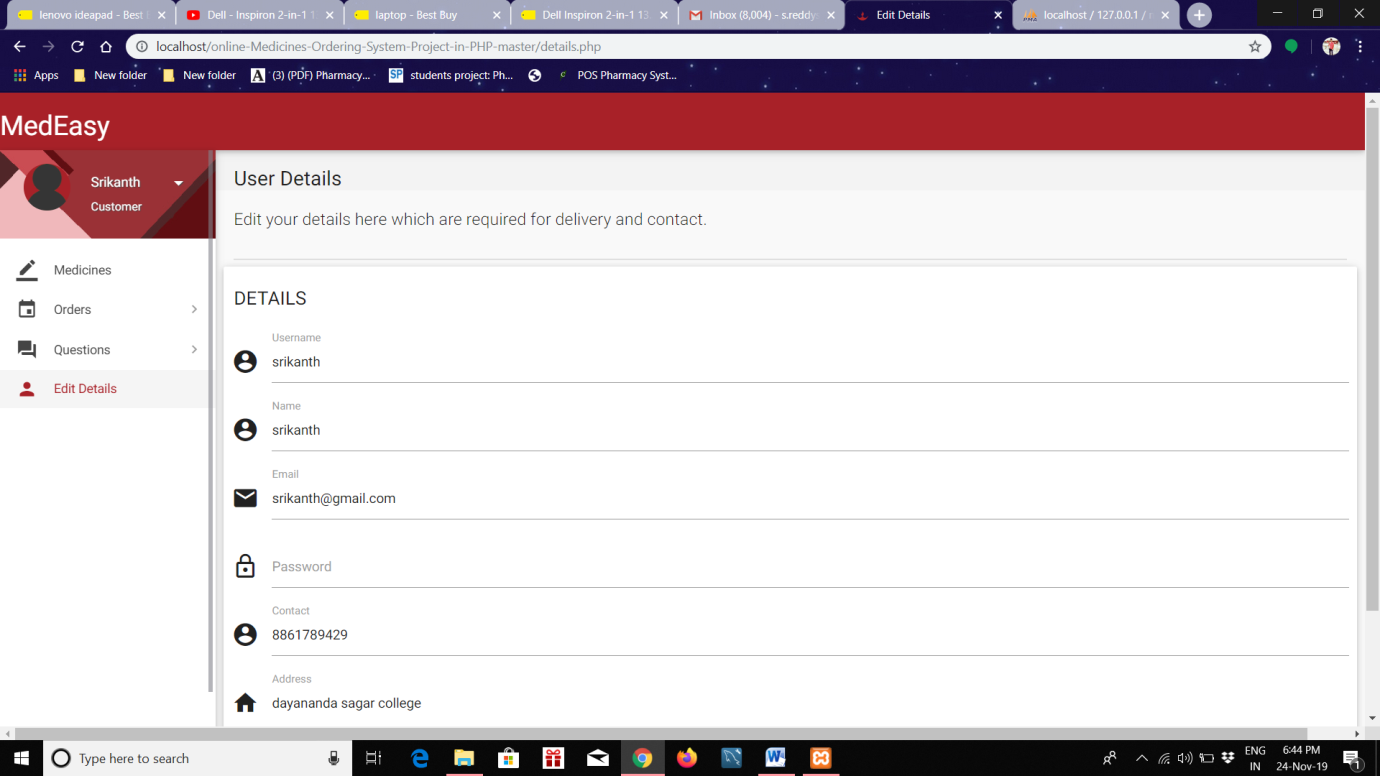
Snapshot 6.7: “Customer orders”

This page will help to check the orders made by customer and can also able to see the status of the order.



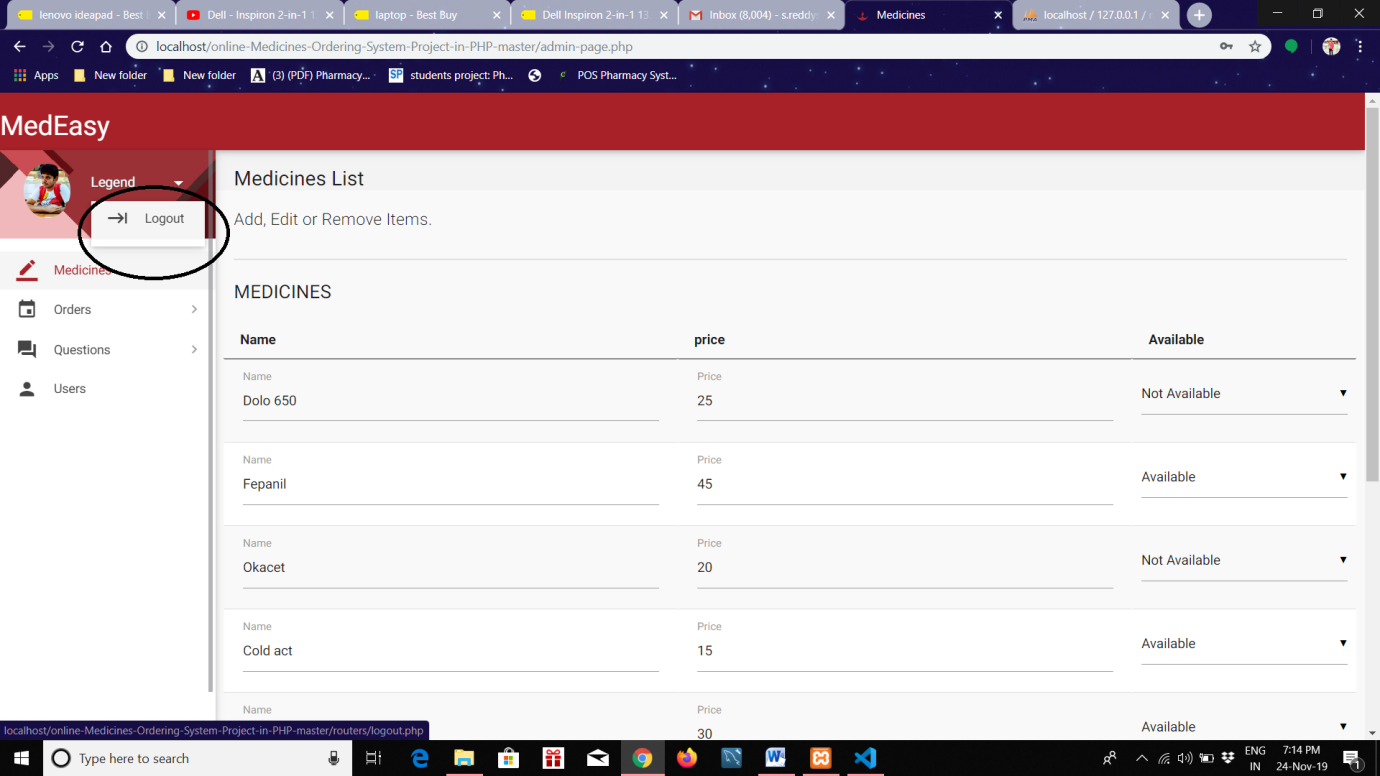
Snapshot 6.8: “Question section”

In this page customer can allowed to ask questions. They can ask any questions .



Snapshot 6.9: “edit details”

This page is used to edit the details of the customer.



Snapshot 6.10: “Log Out”

When you press that logout button, they will be logged out from that page.

**CONCLUSION**

To conclude the description about the project: The project developed using PHP and MYSQL based on the requirement specification of the user and the analysis of the existing system.

Online Medical management system is designed to improve the accuracy, enhance safety and efficiency in the pharmaceutical store. It is a computer based system which helps the Pharmacist to improve inventory management, cost, medical safety etc. Pharmacy management system was developed to ensure the security of information and reliability of Pharmacy records when accessing and providing services to the customers. The information gathered during the data collection was properly analysed and the results provided the basis for the new system. The system was tested and found to be functional and the outputs produced by this system were encouraging. The application will hence reduce the loss of information unlike the existing system and also information will be processed fast.

**FUTURE ENHANCEMENTS**

# We have developed this project for the customers to purchase the medicine through online. This software can be further enhanced to main the track records of sales, purchases, receipts, maintenances and other related issues. It helps in replacing the manual system of record keeping with the modern computerized system. We will message service to inform customers.

**PROBLEM SOLUTION:**

With this application the pharmacist or owner can easily maintain the stock of the medicines. For the customers also they need not to go there and wait to purchase medicines. With this they can order medicines through online in their home itself.

**REFERENCES:**

[1] Matthew MacDonald, "HTML 5 - The Missing Manual", 2nd ed, 2014, O’Reilly.

[2] David Sawyer McFarland, "CSS 3 - The Missing Manual", 3rd ed, 2013, O’Reilly

[3] R. Elmasri and S. Navathe B., “Fundamentals of Database Systems, Global Edition”, Pearson Higher Ed, 2016.

[4] PHP Language Reference

@ <http://php.net/manual/en/langref.php>.

[5] MYSQL Language Reference

@<https://dev.mysql.com/doc/refman/8.0/en/>

[6] HTML Language Reference

@<https://htmlreference.io/>