**CSCE 5430 Deliverables 3**

**Online Pharmacy Portal (Group I)**

**Phase I of development**

**For**

**Online Pharmacy Portal (web page)**

**Version 1.0 (approved)**

**Prepared by: Akella Soujanya**

**Akella Gayatri Sravani**

**Chahnitha Chandu**

**Dodla Nikithanjali**

**University of North Texas, Denton**

**Date: 16th October, 2019**

**Table of contents**

1. Requirements for phase I ……………………………………….. 3
2. UML diagrams ………………………………………………….. 4
3. System Architecture …………………………………………….. 6
4. Test cases ……………………………………………………….. 6
5. Workload division in team ……………………………………… 9
6. User manual …………………………………………………….. 10
7. Product instructions …………………………………………….. 15
8. Peer review and feedback summary ……………………………. 15
9. **Requirements for Phase I**

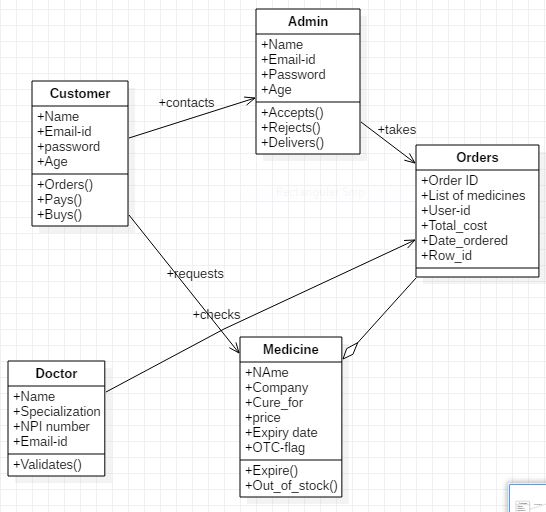
In the first phase of development, we have implemented a few requirements for the webpage “Med Anytime”, an online pharmacy portal where a person can order medicines and few other similar products online.

We are specifying the requirements implemented in phase I below:

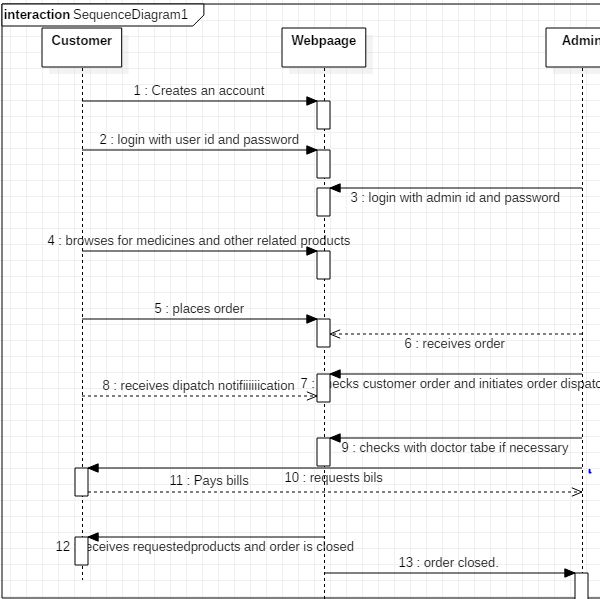
(Requirements were not properly specified in Deliverables 2 as per feedback.)

* All the basic pages required for the page will be ready for phase 1 of development
* User should be able to login to their account securely on providing the correct username (email-id) and password.
* An error prompt must be given in case of incorrect password.
* User should be able to register as a new user.
* User should be able to view featured products available for purchase after login.
* Database connectivity is made with MedAnytime. Database consists of tables for user accounts details, products sold on the web page. (More tables will be added in future.)
* Admin should be able to view medicines in database after login.
* Admin should be able to update the products in database, be it price or quantity or expiry date.
* Admin must be able add or delete products in the inventory and these changes should be reflected on the user page appropriately.

1. **UML Diagrams**
   1. **Class Diagram**

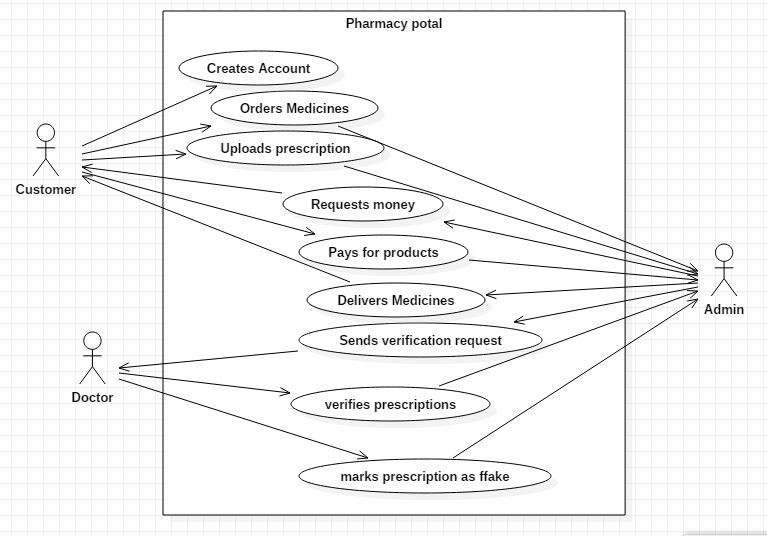
****

* 1. **Sequence diagram**

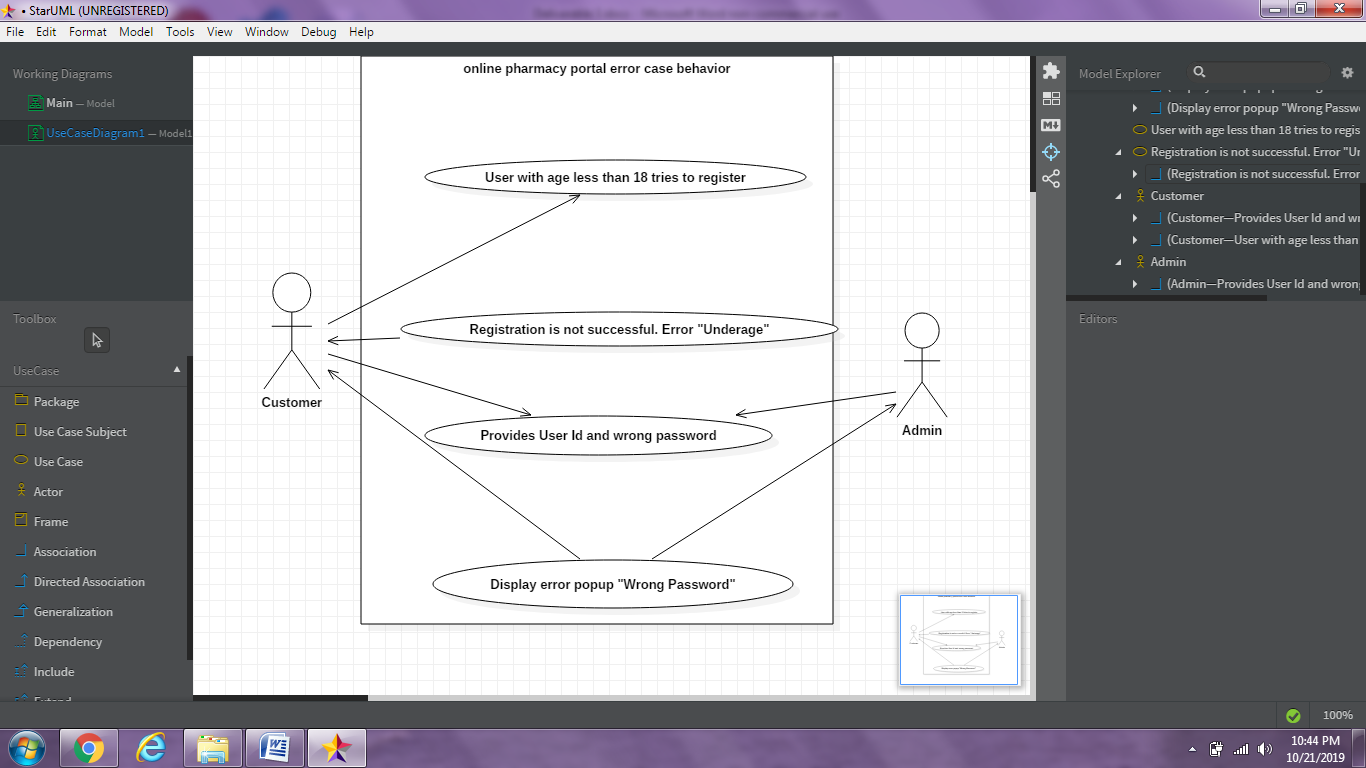
****

* 1. **Usecase Diagram**

1. **Normal case**



1. **Error case** (In this phase of development we consider two error cases, one during registration and another during user or admin login)

****

1. **System Architecture**

**(Executes php server)**

**Online Pharmacy portal**

Web Browser

**Request Response**

**XAMPP**

**Database (phpmyAdmin)**

1. **Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S. No** | **Requirement** | **Test Priority** | **Test Steps & pre-conditions** | **Expected Result** | **Obtained Result** | **Success/ Failure** | **Name of Tester** |
| **1** | Successfully launch the login page. | High | For the login page to be launched, it needs to satisfy all the pre-requisites. Next type the URL for the webpage. | The login page must be displayed. | The login page has been displayed. | Success | Nikitha |
| **2** | User should go to register page when they click on register. | High | User should just click on the **Register** button. | Registration page must be displayed | The registration form is successfully displayed. | Success | Soujanya |
| **3** | User should be registered if the email-id provided is not already in Database | High | The user should ensure they haven’t already created account with same mail. | Successful user registration | User now has an account in MedAnytime | Success | Nikitha |
| **4** | Registration in case the email-id is already in the database | Medium | An already existing user should attempt to register again. | User registration unsuccessful. Account already exists. | Registration page threw an error. | Success | Chahnitha |
| **5** | Registration by user who is less 18 years old | Low | A user who is trying to register enters date of birth in 2003. | User registration should not be successful. | Error: “**Age should not be lass than 18**” | Success | Soujanya |
| **5** | On providing correct details like email-id and password in login page | High | User should be registered. Enter login credentials, keeping in mind that they can be case-sensitive. | User page must be opened. | User page is launched | Success | Soujanya |
| **6** | On providing wrong email-id. | Medium | User should enter email-id with some change, like ‘Some’ instead of ‘same’ | Page should display, invalid user message. | Error: “**Not a registered user**” | Success | Gayatri |
| **7** | On providing valid email-id but wrong password. | High | User should enter their correct email-id and wrong password | Page should display a warning that says, Wrong password | Error: “**Wrong Password**” | Success | Gayatri |
| **8** | On providing correct admin details and clicking login. | High | Admin must enter the same credentials used at the time of creation | Admin page must be launched. | Admin home page is launched | Success | Chahnitha |
| **9** | On the user page… | Medium | User must scroll up and down on their home page. | Featured products must be displayed. | Featured products are displayed. User can click on the products to view their details. | Success | Gayatri |
| **10** | On the admin page… | Medium | Admin must click on inventory. | Product inventory must be displayed. | Product inventory is displayed. It shows a list of all products sold in the page with quantity and expiry date. | Success | Soujanya |
| **11** | Add/delete products on admin page | High | Admin must click on ‘**add product’** button in the bottom of inventory list. Enter all necessary details about the product. Similarly | The product should be added into the list of products in the inventory. | Product was successfully added to list of products. | Success | Nikitha |
| **12** | Changes made in the admin page should be reflected on the user page. | High | Make changes to any product on the admin page. | The same change must be reflected on user page | The changes were visible on the user page. | Success | Chahnitha |

**Test cases (Unit Testing)**

For the unit testing, we use PHPUnit. PHPUnit is a unit testing framework for the PHP programming language. It is an instance of the xUnit architecture for unit testing frameworks that originated with SUnit and became popular with JUnit. PHPUnit was created by Sebastian Bergmann and its development is hosted on GitHub.

**Registrationtest.php**

<?php

require\_once('registration.php');

class TestRegister extends PHPUnit\_Framework\_TestCase

{

public function tearDown(){ }

protected function setUp()

{

parent::setUp();

$\_POST = array();

}

public function testRegisterIsValid()

{

$POST = array( 'username' => 'test@gmail.com','password' => 'password','firstname' => 'test','lastname' => 'test', 'gender' => 'Male', 'DOB' => '1994-07-13', 'terms' => 'chkBox', 'govtID' => 'test.jpg' );

$user = new register($POST);

$user->save();

}

}

?>

**LoginTest.php**

<?php

require\_once('login.php');

class Testlogin extends PHPUnit\_Framework\_TestCase

{

public function tearDown(){ }

protected function setUp()

{

parent::setUp();

$\_POST = array();

}

public function testLoginIsValid()

{

$\_SERVER["REQUEST\_METHOD"] = 'POST';

$connObj = new login();

$\_POST['username'] = 'sravani95@gmail.com';

$\_POST['password'] = 'gaya3';

$this->assertNotNull($connObj>login($\_POST['username'],$\_POST['password'], $\_SERVER["REQUEST\_METHOD"]));

}

}

?>

1. **Workload Division**

The work was equally divided among all group members. I am mentioning below a list of all the jobs that have been completed so far along with the name of the person who has completed it.

* Development of the login page (By Soujanya)
* Creation and setting up of Database (By Gayatri)
* Configuring Xampp server (By Chahnitha)
* Development of the user page (By Nikitha)
* Development of the admin page (By Chahnitha)
* Development of the Registration page ( By Soujanya)
* Development of the inventory page (By Gayatri)
* Development of products display in user page (By Nikitha)
* Validations in registration page to check age constraint (By Soujanya)
* Connection of each of the above mentioned pages appropriately (By Chahnitha)
* Admin account and database manager (By Gayatri)
* Add/ delete products in the database through admin account (By Soujanya)
* Updating of database Whenever admin makes changes in admin pages (By Nikitha)
* Being able to view product details for each product displayed in user page or admin page (By Chahnitha)
* Adding all products, and images in the page (By Soujanya, Nikitha)
* Documentation (is done by everyone)
* Project repository manager (Gayatri)

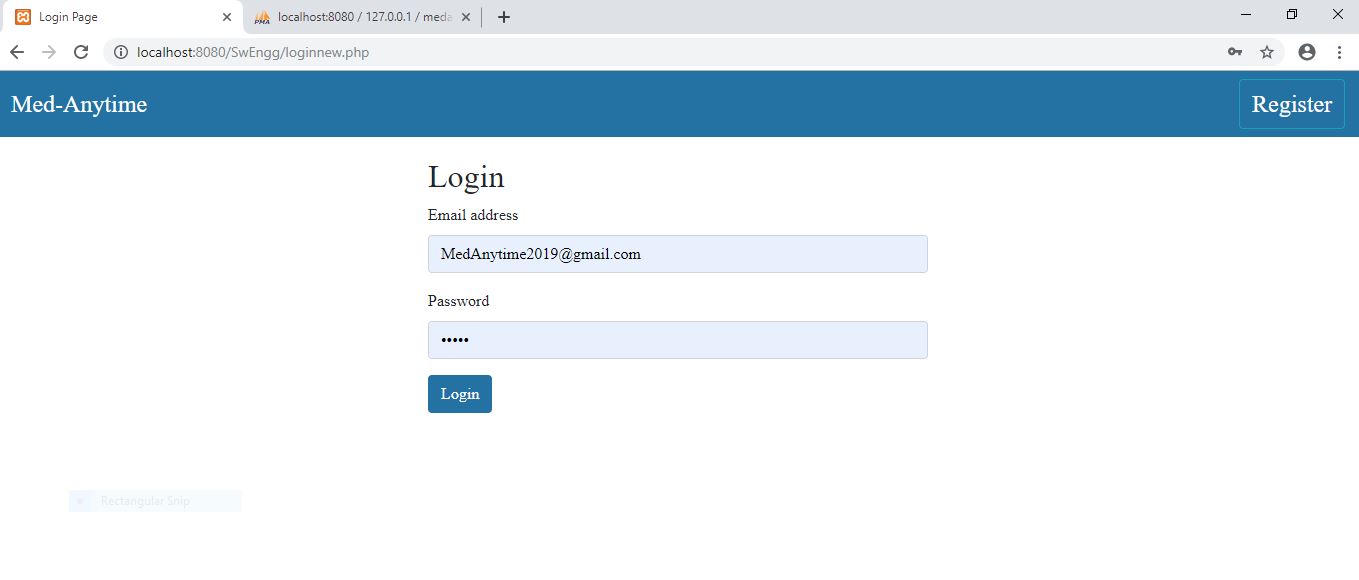
1. **User Manual & usage instructions**

This web page has been designed so that it can be used by anyone with utmost ease. Since this is not an application that needs to be installed and run in our system, the process is so much easier.

To be able to use this webpage, a person needs to be fulfilling a few pre-requisites like:

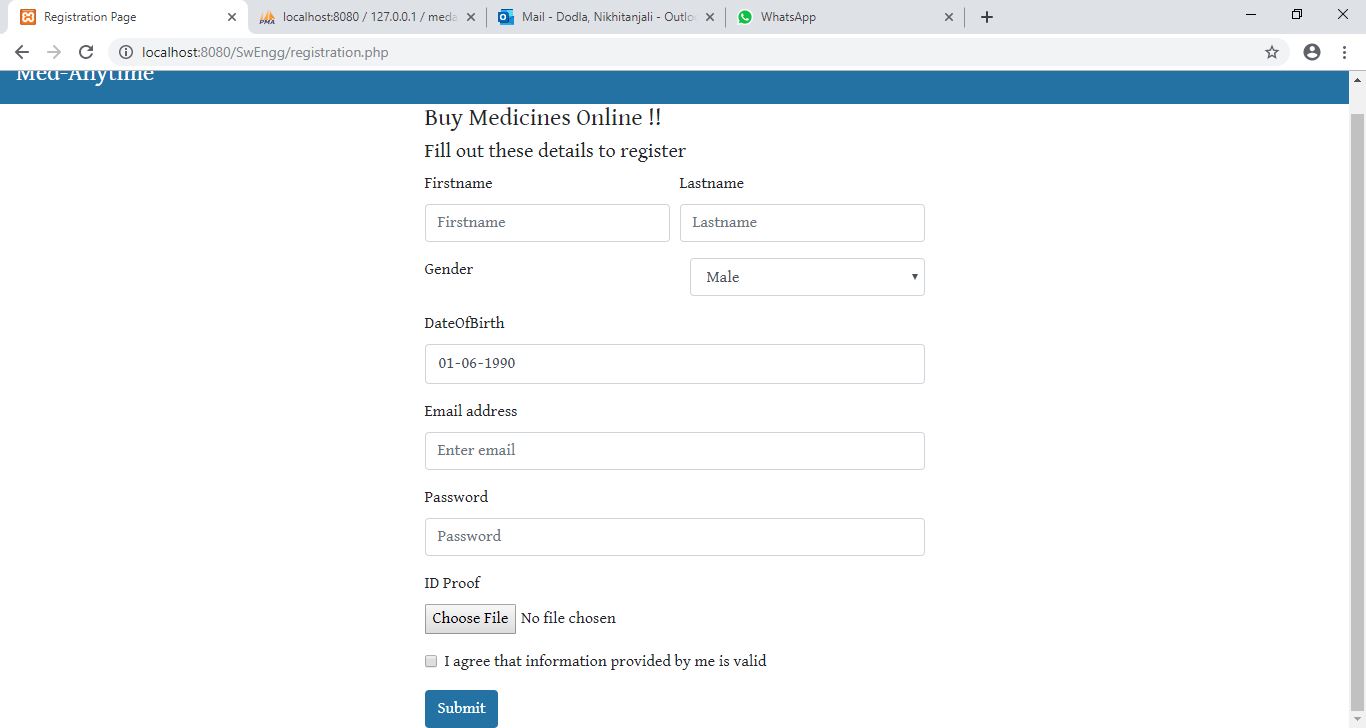
* Must be 18 years or older
* Must have a computer/laptop that is run on windows operating system (version 7 or more)
* Should have a valid ID proof and a personal email ID

To begin with, the customer or user has to open their web browser (preferably chrome) and open the webpage for **MedAnytime**. The login page looks just like the screenshot below.

****

If you are a new user, click on the **register** button on the right top corner. It takes you to a different page. This page is basically just like a form that needs to be filled up. The form asks for a few details like name of the user, email id, gender and ID proof. After filling up all of this information in the required format, the user will have to check a box, which basically means he/she accepts that all the information provided by them is fully correct to the best of their knowledge.

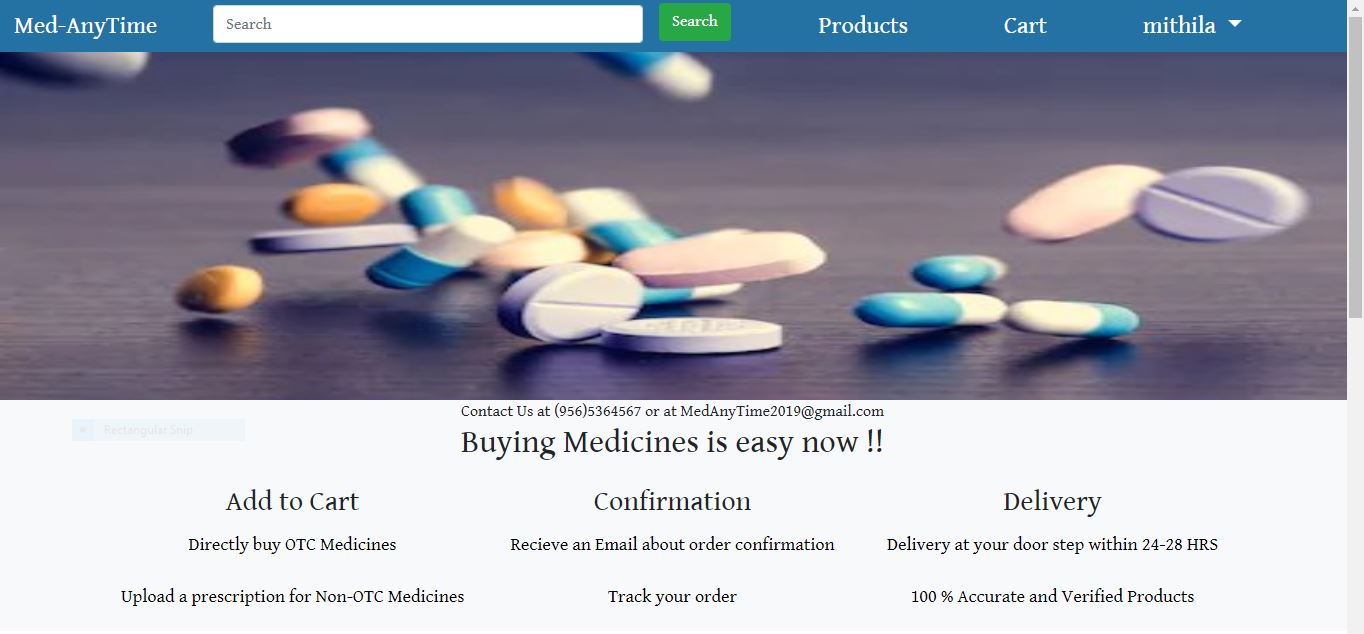
The registration page looks as shown below.

****

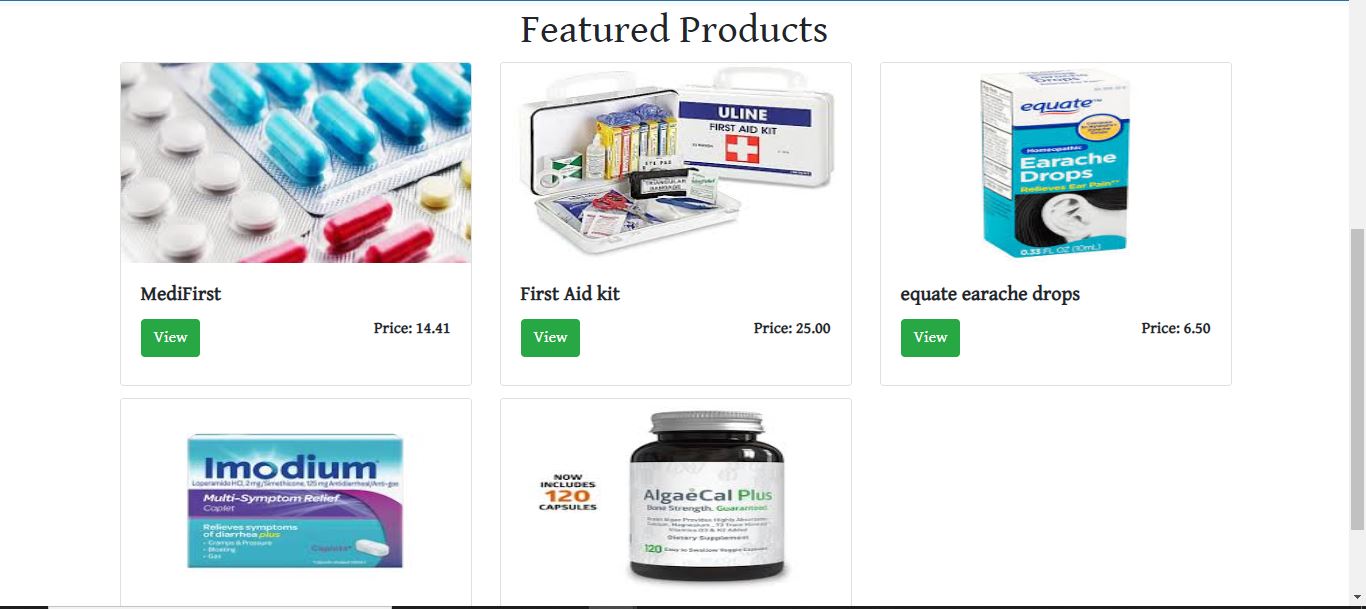
After registration, the user is automatically re-directed to the home page again.

Now, if you’re a returning user or you have just registered, the user has to login into their personal account. For safety sakes, MedAnytime does not allow unregistered users to access any of its pages.

To login, one needs to enter their email-id (provided while registering) and password (same as the password given at registration). After this, click on the **Login** button. This will take you to the user’s home page. It looks as shown below. (Shown in the below image is a snapshot of the user home page. Home page belongs to a user named Mithila.)

****

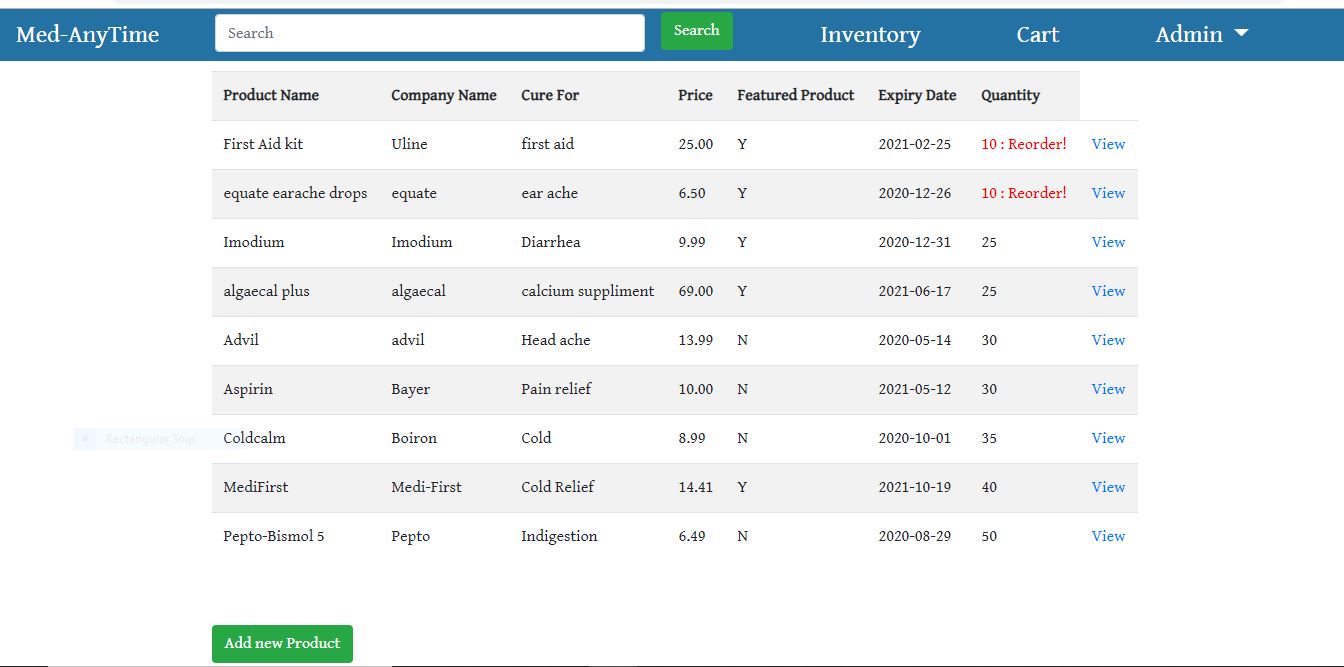
The user page also displays a list of featured products that can be bought on MedAnytime. That can be seen in the screenshot below.

****

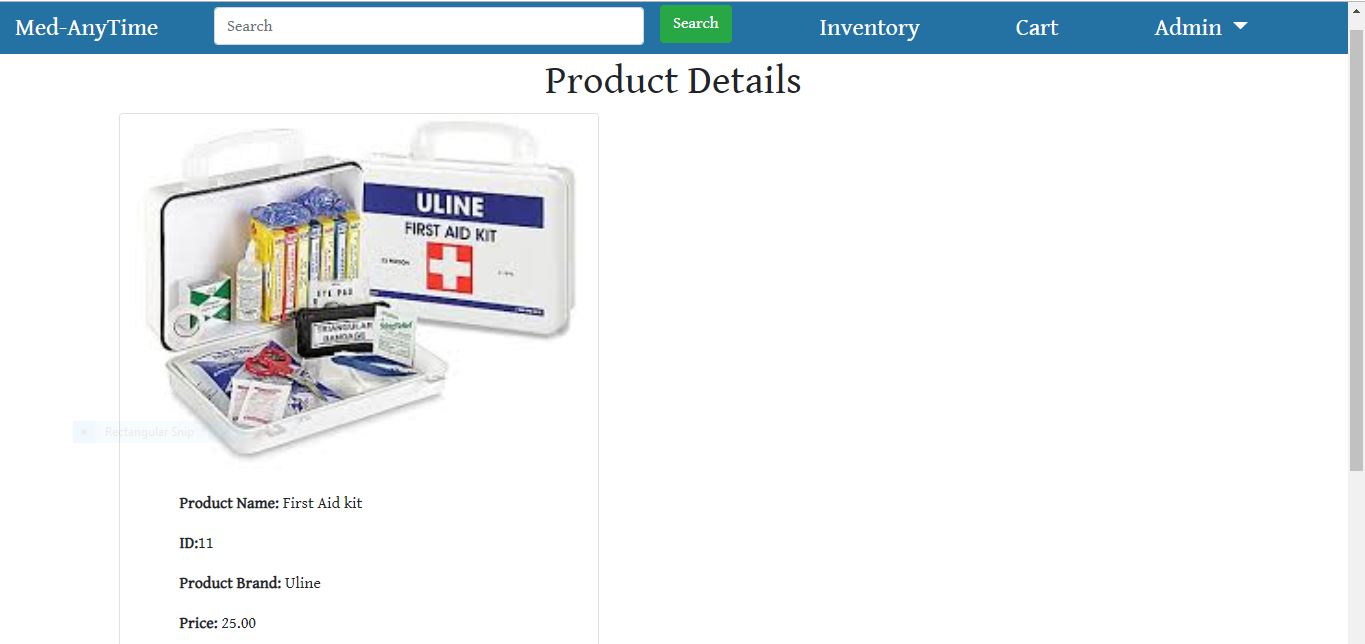
Admin login also is pretty similar. There is no registration option for admin as the page can only have one admin and was created manually. The admin just has to enter his/her email-id and password and the page will take them to the admin home page instead of the user homepage.

On the respective home page of the user, they can perform a lot of activities most of whose functionality is not fully developed yet. So, the manual for the first phase of development ends here for the user.

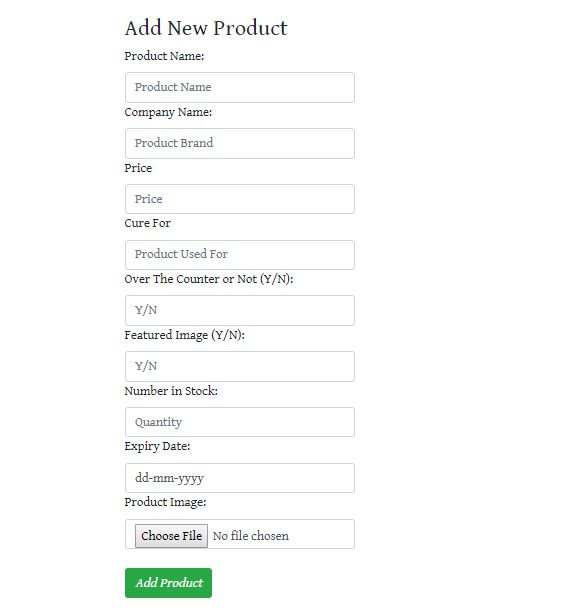
The admin on the other hand can view all the products in the inventory by clicking on the **Inventory** button on the top bar in his home page. The list of products will be displayed as shown below.



If the admin clicks on the **view** button beside any of these products, he can get details about that product and also, will get an option to edit those details on this page. Given below is a snapshot of what the details would look like for a first aid kit.



Also, on the inventory page, the admin can see a **Add new Product** button. On clicking this button, admin can add more products to the webpage. Given below is the snapshot of all the details needed to add a new product to the page.



There is further functionality that is to be added to the admin page as well. But for now, this is all we have for the admin. All other functionalities will be developed in the fore coming development phases.

Thank you for taking your time to read it. Any queries, questions or feedback can be mailed to [**MedAnytime2019@gmail.com**](mailto:MedAnytime2019@gmail.com)**.**

**Installation Instructions:**

The installation process for this webpage requires us to have some software on the system in which we want to run the page on. Here, we make an assumption that the user has already downloaded the project folder onto his/her system before starting the installation process.

* Firstly, we need a working version of the XAMPP control panel (we have used XAMPP 3.2.4).
* Next we need to start apache service and the MySql service in the XAMPP control panel.
* Please place the downloaded project folder in the following folder on your system.

**C: /xampp/htdocs**

* Now, open a compatible web browser, preferably Google chrome. Now type the url: localhost:**port\_number/SwEngg/loginnewphp**
* Now, if all the components are working correctly, the login page should be displayed. This means, the webpage is now ready and available for use.
* If not, the user might want to check his mysql connection or any other necessary component that might not be running correctly. (In most cases, it is displayed on the browser itself that some component is not working correctly).

1. **Peer review and feedback**

In the previous peer review session, we received the following feedback and suggestions:

* Including system architecture would be more helpful than having ER diagrams in the SRS document.
* They felt that our Development plan phases are not clearly explained.

For the system architecture part, we have specified the architecture and overall structure in deliverables 3 document. We have discussed the phases of development phases with our peers and clarified all their questions. Any other minor ambiguities were also discussed.