CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The purpose of any e-commerce website is to help customers narrow down their broad ideas and enable them to finalize the products they want to purchase. As the customer selects more and more features or options from the facets provided, the search narrows down to a small list of products that suits their choice. If the list is small enough and the customer likes one of listed products, he or she will make the purchase. The website stores information of the items and the categories to which they belong to. It also stores admin and user details. The website maintains two levels of users: administer level and user level. The administrator will be able to enter all the details into the website. The user can search for items and place their orders

1.2 PROBLEM STATEMENT

In this day of internet services, a provision for online services for the Elderly's care is scarce. Therefore, there is a need for a system that supports and provides a platform for the elderly's care and shopping of their needs to enable the day-to-day sales functions. The system allows customers to order products and services using this web project. The demographic of target are the elderlies of the society. The system enables vendors to set up their products online, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories.

1.3 SCOPE

The agenda is to design an online shopping site that manages the products and simplifies customer's purchases without having to visit stores physically. The outlet will use the internet as the sole method for selling goods to its consumers at the lowest found prices.

CHAPTER 2

SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- Processor : Intel core i3 and above version
- Speed : 1.20GHz
- RAM : 2GB and above
- Keyboard : Standard Windows Keyboard
- Mouse : Two or Three Button Mouse
- Monitor : SVGA

2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 8 and above
- Frontend Tool : NetBeans
- Server : Xampp Server
- Backend Layer/Database : SQLite
- Frontend : HTML, CSS, Bootstrap 3
- Backend : PHP

CHAPTER 3

DESIGN

3.1 SYSTEM DESIGN

The aim of the site is to present the following factors to the customers for better usability:

- Returning to different parts of the site after adding an item to the shopping cart.
- Easy scanning and selecting items in a list.
- Effective categorical organization of products.
- Simple navigation from home page to information and order links for specific products.
- Obvious shopping links or buttons.
- Minimal and effective security notifications or messages.
- Consistent layout of product information.

3.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

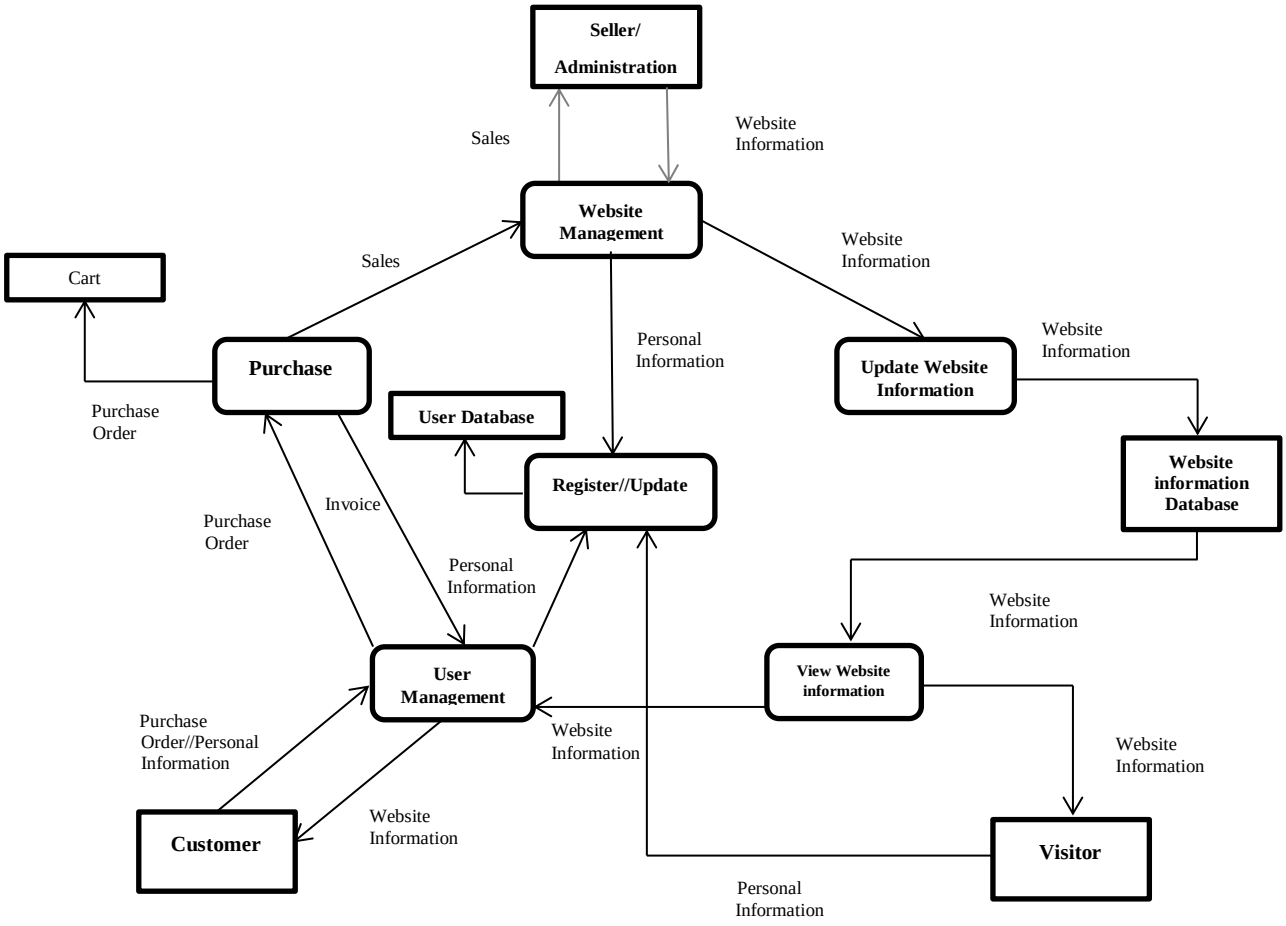

Figure 3.1 Data flow diagram

CHAPTER 4

IMPLEMENTATION

4.1 INTRODUCTION

The Internet has become a major resource for several websites. Among many websites such as blogs, personal websites and informational websites, e-commerce websites are very popular. The ecommerce websites have gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds.

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process. Although the issue of security remains the primary reasons why more people do not purchase terms online, more and more users can be expected to frequently purchase items online because of their increasing confidence in current encryption technologies.

A store is a virtual store on the Internet where customers can browse the catalogue and select items of interest. The selected items may be collected in a shopping cart and then confirmed for purchasing. The objective of the system is to develop an e-store, where items required by targeted demographic can be bought from the comfort of home through the Internet.

The system is implemented using a 3-tier approach, with a backend database, a middle tier of XAMPP server and PHP, and a web browser as the front-end client. In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client-side scripting techniques, programming languages such as PHP, JavaScript, HTML, Bootstrap and CSS for styling, relational databases such as MySQL.

- **HYPER TEXT MARKUP LANGUAGE [HTML]**

  HTML was specifically developed to use along with the Hyper Text Transfer Protocol (HTTP) to encode documents for display on the World Wide Web. HTML is defined in the HTML Standard, currently Version 4.0x.HTML standards are recommended by the World Wide Web Consortium, W3C. W3C also oversees the standardization of technologies related to the World Wide Web and publishes the HTTP (Hypertext Transfer Protocol) standards. HTML is initials for Hyper Text Markup Language.

- **CASCADING STYLE SHEET [CSS]**

  CSS is the language used for describing the presentation of web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.CSS is independent of HTML and can be used with any XML-based mark-up language. There are three types of CSS they are:

  - Inline CSS
  - Internal CSS
  - External CSS

  ➢ **Inline CSS:**

  Inline CSS refers to CSS found in an HTML file. It is found in the head of a document between style tags. Inline CSS simply means putting your CSS into your HTML file instead of an external CSS file.

  ➢ **Internal CSS:**

  Cascading Style sheets (CSS) are files with styling rules that govern how your website is presented on screen. CSS rules can be applied to our website's HTML files in various ways. We can use an external style-sheet, an internal style-sheet, or an inline style. Each method has advantages that suit particular uses.

  ➢ **External CSS:**

  An external style sheet is a separate file linked to an HTML web page. It comes with a **.CSS** filename extension. All the styles that need to be used on a website can be declared in an external style sheet.

- **PHP**

  PHP stands for Hypertext Pre-processor. It is an open source, server-side, scripting language used for the development of the web applications. By scripting language, we mean a program that is script-based (lines of codes) written for the automation of tasks. PHP is a script language and interpreter that is freely available and used primarily on Linux Web servers. PHP, originally derived from Personal Home Page tools, now stands for PHP: Hypertext Pre-processor. Some features of PHP are: simple, interpreted, faster, open source, platform independent, case sensitive, error reporting. The PHP script can be placed anywhere in the document. The default file extension for PHP files is ".php". A PHP file normally contains HTML tags, and some PHP scripting code.

- **BOOTSTRAP**

  Bootstrap (front-end framework) is a free and open-source front-end framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It is a framework that helps design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, image carousels, etc. It also gives us support for JavaScript plugins. Bootstrap was also programmed to support HTML5 and CSS3. Some important features of bootstrap are: Easy to begin with, easily customizable, responsive utility classes, drop-down component menu, bootstrap templates etc.

- **XAMPP**

  XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming language. XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer. With the advantage of common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami. XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also comes with a number of other modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version).

4.2 ALGORITHM/TECHNIQUE USED

CREATE AND LOGIN:

The create feature allows user to create a new user. The same username and password is used to login If the user is authenticated the test results are inverted to the record file and index file. If he is not authenticated, it is not written to the file.

Step 1: Start

Step 2: User enters credentials.

Step 3: Check if the credentials match

If yes, login successful

Step 4: If credentials do not match,

New user is created. Write the credentials to the data file

Step 5: Stop

ADD TO CART:

Step 1: Select an item

Step 2: Check for user login

If yes, proceed

Else, login or Signup

Step 3: Select the count

Step 4: Now the item is added to Cart

Step 5: Check-out.

4.3 CLASSES AND METHODS

LOGIN/SIGN UP:

```
<?php

require("includes/common.php");

$email = $_POST['e-mail'];

$email = mysqli_real_escape_string($con, $email);

$password = $_POST['password'];

$password = mysqli_real_escape_string($con, $password);

$password = MD5($password);

// Query checks if the email and password are present in the database.

$query = "SELECT id, email FROM users WHERE email='" . $email . "' AND password='" .
$password . "'";

$result = mysqli_query($con, $query)or die($mysqli_error($con));

$num = mysqli_num_rows($result);

// If the email and password are not present in the database, the mysqli_num_rows returns 0, it is
assigned to $num.

if ($num == 0) {

  /*$error = "<span class='red'>Please enter correct E-mail id and Password</span>";*/

  header('location: login.php?error=' . $error);

} else {

  $row = mysqli_fetch_array($result);

  $_SESSION['email'] = $row['email'];

  $_SESSION['user_id'] = $row['id'];

  header('location: index.php');

  }
```

SIGN-UP

```
<?php
require("includes/common.php");
  // Getting the values from the signup page using $_POST[] and cleaning the data submitted by the user.
  $name = $_POST['name'];
  $name = mysqli_real_escape_string($con, $name);
  $email = $_POST['e-mail'];
  $email = mysqli_real_escape_string($con, $email);
  $password = $_POST['password'];
  $password = mysqli_real_escape_string($con, $password);
  $password = MD5($password);
  $contact = $_POST['contact'];
  $contact = mysqli_real_escape_string($con, $contact);
  $city = $_POST['city'];
  $city = mysqli_real_escape_string($con, $city);
  $address = $_POST['address'];
  $address = mysqli_real_escape_string($con, $address);
  $regex_email = "/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})$/";
  $regex_num = "/^[789][0-9]{9}$/";
  $query = "SELECT * FROM users WHERE email='$email'";
  $result = mysqli_query($con, $query)or die($mysqli_error($con));
  $num = mysqli_num_rows($result);
  if ($num != 0) {
    $m = "<span class='red'>Email Already Exists</span>";
    header('location: signup.php?m1=' . $m);
  } else if (!preg_match($regex_email, $email)) {
```

```
    $m = "<span class='red'>Not a valid Email Id</span>";
    header('location: signup.php?m1=' . $m);
  } else if (!preg_match($regex_num, $contact)) {
    $m = "<span class='red'>Not a valid phone number</span>";
    header('location: signup.php?m2=' . $m);
  } else {
    $query = "INSERT INTO users(name, email, password, contact, city, address)VALUES('" .
$name . "','" . $email . "','" . $password . "','" . $contact . "','" . $city . "','" . $address . "')";
    mysqli_query($con, $query) or die(mysqli_error($con));
    $user_id = mysqli_insert_id($con);
    $_SESSION['email'] = $email;
    $_SESSION['user_id'] = $user_id;
    header('location: products.php');
  }
?>
```

PREFERENCES:

For logged out users, we will show button with Add to cart value with href="login.php".

For logged in users, button will be disabled if a particular item is already added to cart by the user. Else show the link to add the item to the cart.

```
<?php if (!isset($_SESSION['email'])) { ?>
<p><a href="login.php" role="button" class="btn btn-primary btn-block">Buy Now</a></p>
<?php
} else {
//We have created a function to check whether this particular product is added to cart or not.
if (check_if_added_to_cart(1)) { //This is same as if(check_if_added_to_cart != 0)
echo '<a href="#" class="btn btn-block btn-success" disabled>Added to cart</a>';
} else {
?>
```

<a href="cart-add.php?id=1" name="add" value="add" class="btn btn-block btn-primary">Add to cart</a>

<?php

} } ?>

ADD TO/VIEW CART:

Write the insert query to add items id (from url) and user id from the session and status would be Added to cart.

INSERT INTO user_items(user_id, item_id, status) VALUES('$user_id', '$item_id', 'Added to cart')

<?php

require("includes/common.php");

if (isset($_GET['id']) && is_numeric($_GET['id'])) {

$item_id = $_GET['id'];

$user_id = $_SESSION['user_id'];

$query = "INSERT INTO users_items(user_id, item_id, status) VALUES('$user_id', '$item_id', 'Added to cart')";

mysqli_query($con, $query) or die(mysqli_error($con));

header('Location:'.$_SERVER['HTTP_REFERER']);}

?>

Check-if-added.php inside includes folder

● Make a function check_if_added_to_cart($item_id). This function will get user_id from the session.

● Include common.php file inside the function.

● This function will take user id and items id and will check whether there is an entry in users_products table with status 'Added to cart'.

SELECT * FROM user_items WHERE item_id='$item_id' AND user_id ='$user_id' and status='Added to cart'

4.4 MODULE DESCRIPTION

Assistance of Golden-Ager consists of four modules. Items are inserted in items table of database. Operations involved in these four modules are:

1. **USER LOGIN/SIGNUP:**

   The user has an existing account, he/she is prompted to login. If it is a new user, he/she will have to create an account using the sign up module to buy items and add it to the cart. Login Module is a portal module that allows users to type a user name and password to login. This module can be placed on any module tab to allow users to login to the system. User should fill require fields to create account and same to login.

2. **PREFERENCES:**

   This module categorises the available products and displays the categories for ease of purchase by the user. The products are sorted individually here naming a category each and has products.

3. **CART:**

   Cart consists of products selected by the user for purchase. It allows of confirmation/removal of items. The cart is empty when the product is not selected. The products can be added only once.

4. **ABOUT US:**

   It consists of information of the students involved in the development of the platform and information about the e-commerce shopping store. The importance of the website over the other shopping websites and stores around the world.

CHAPTER 5

SAMPLE OUTPUT

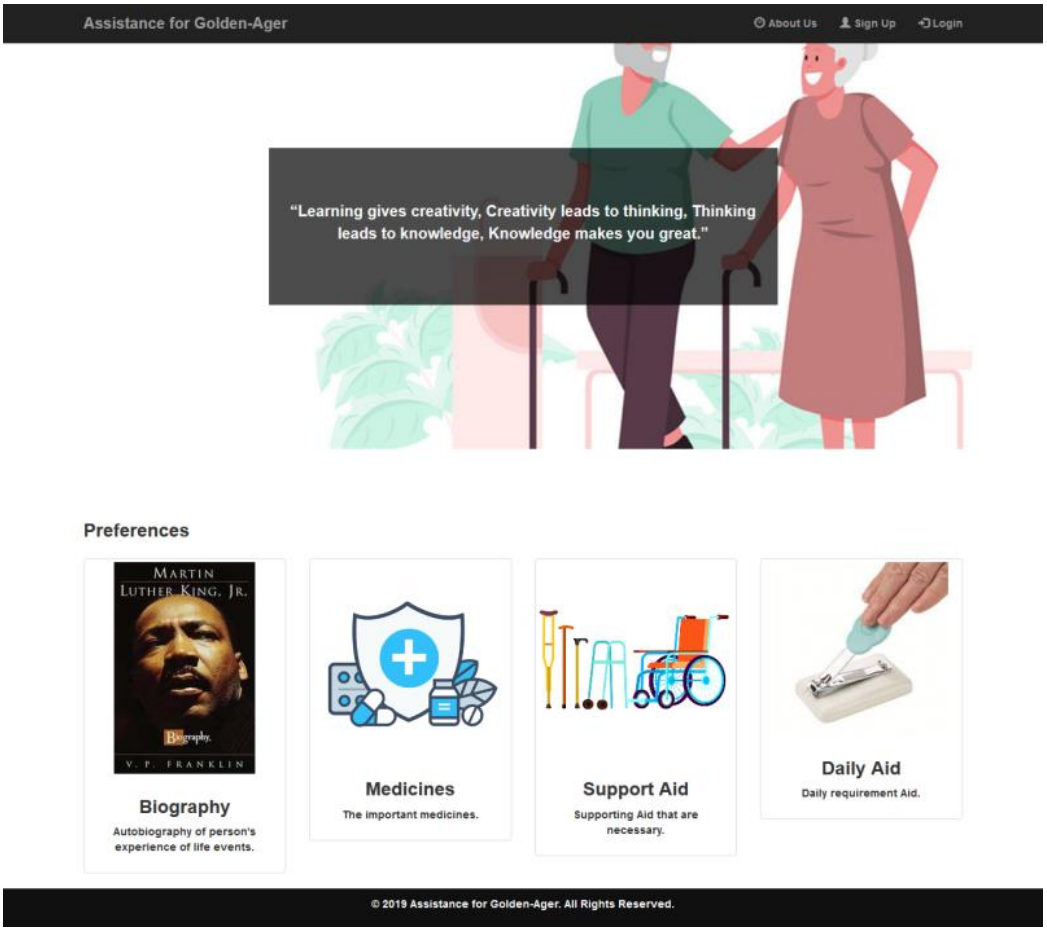

Fig 5.1: Homepage.

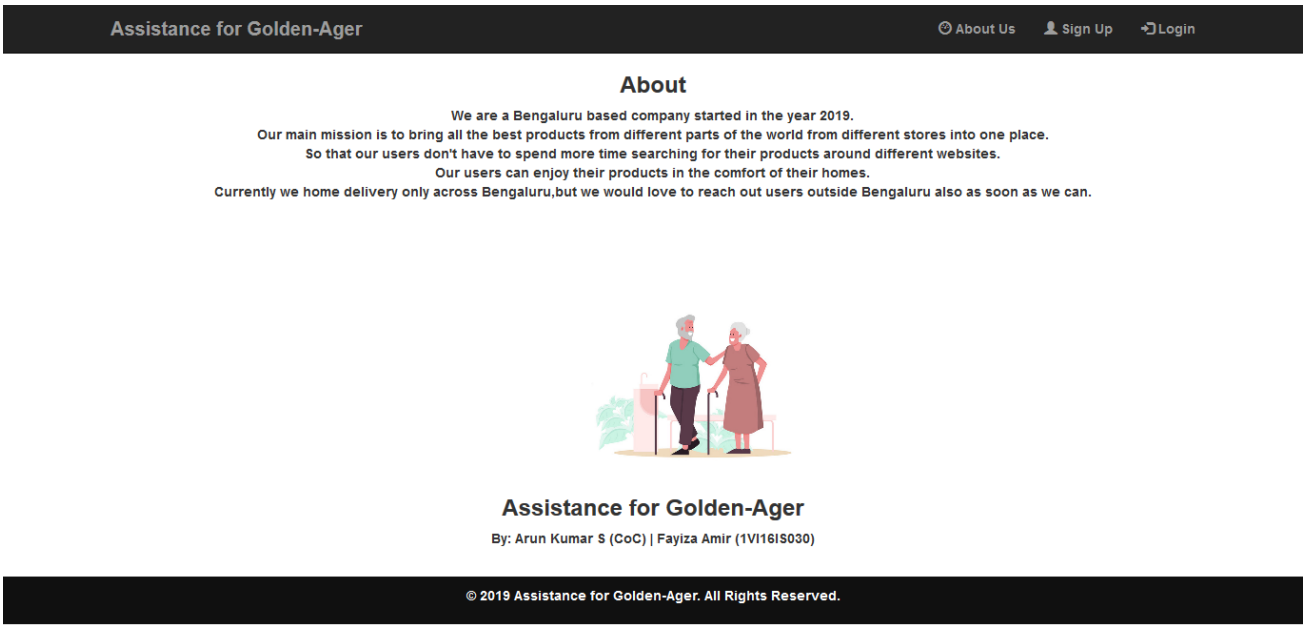

Fig 5.2: About Us.

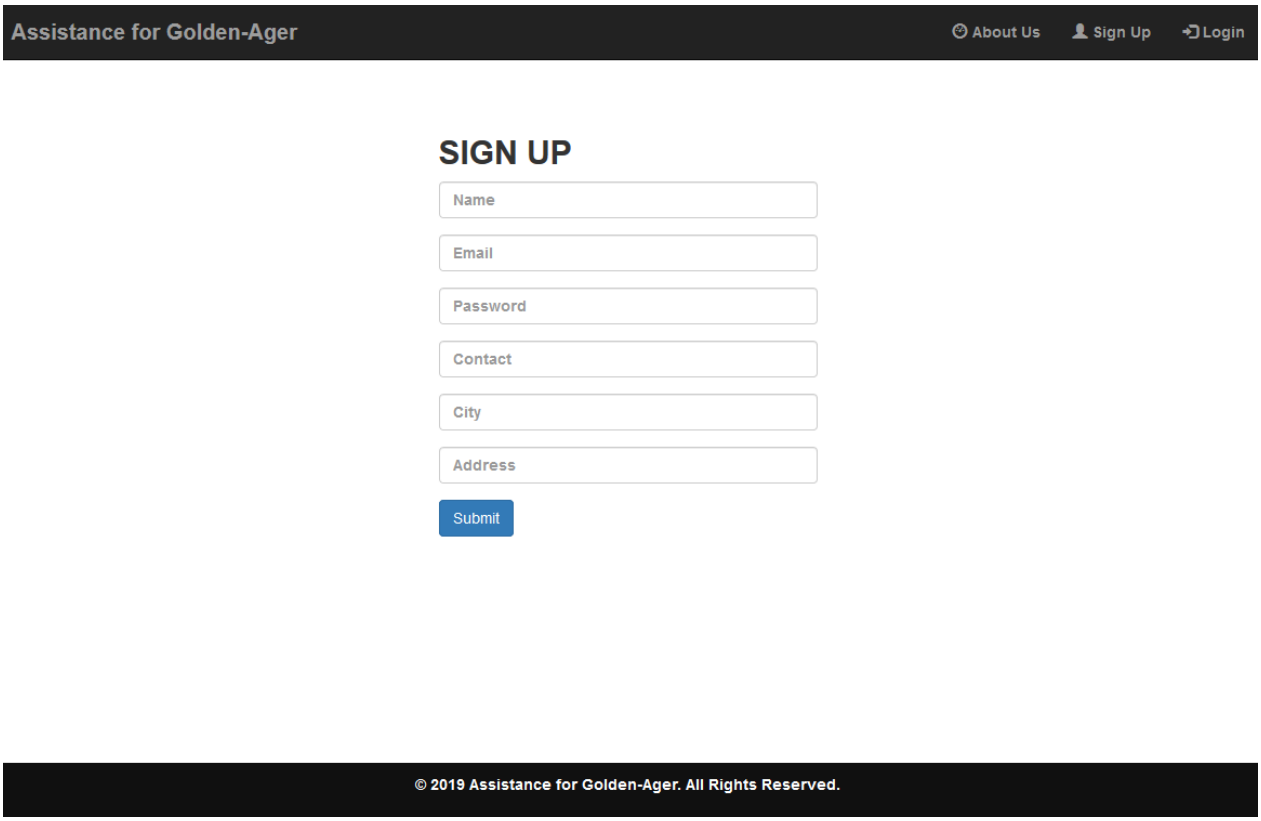

Fig 5.3: Sign-up new user.

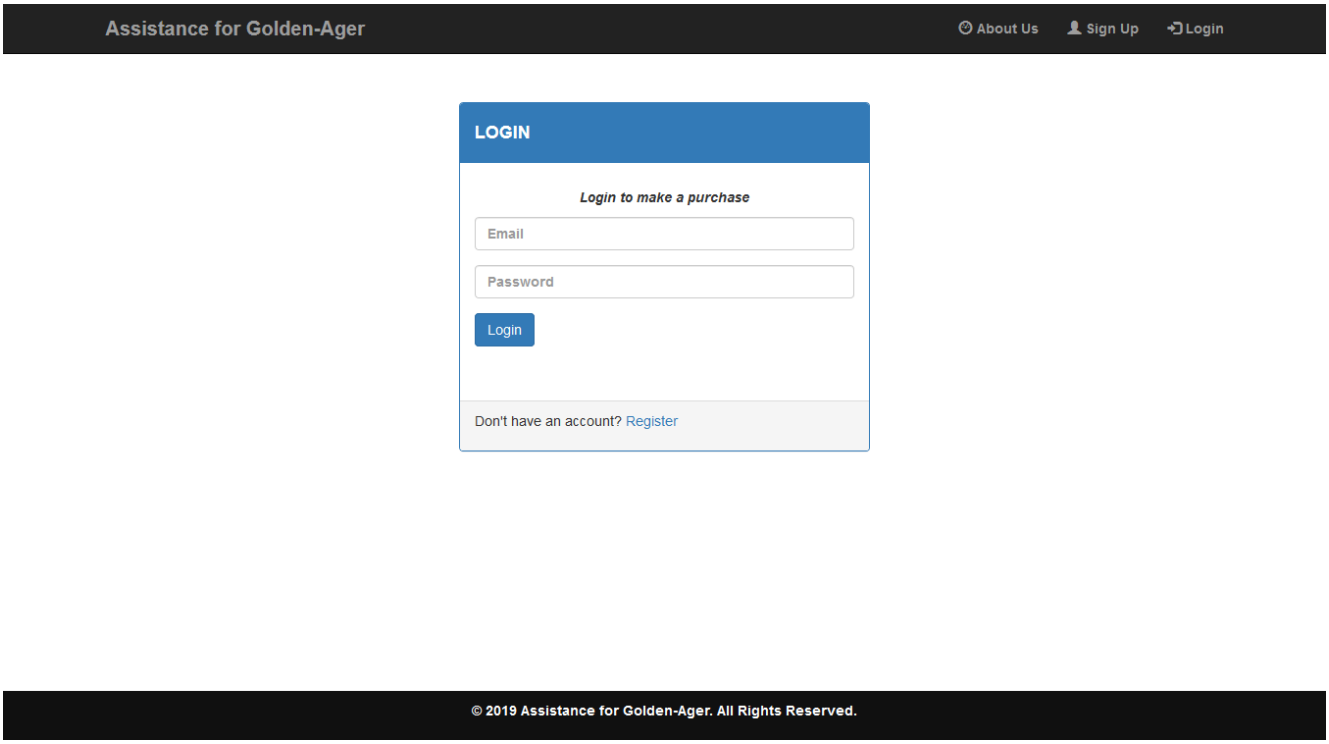

Fig 5.4: Existing user login.

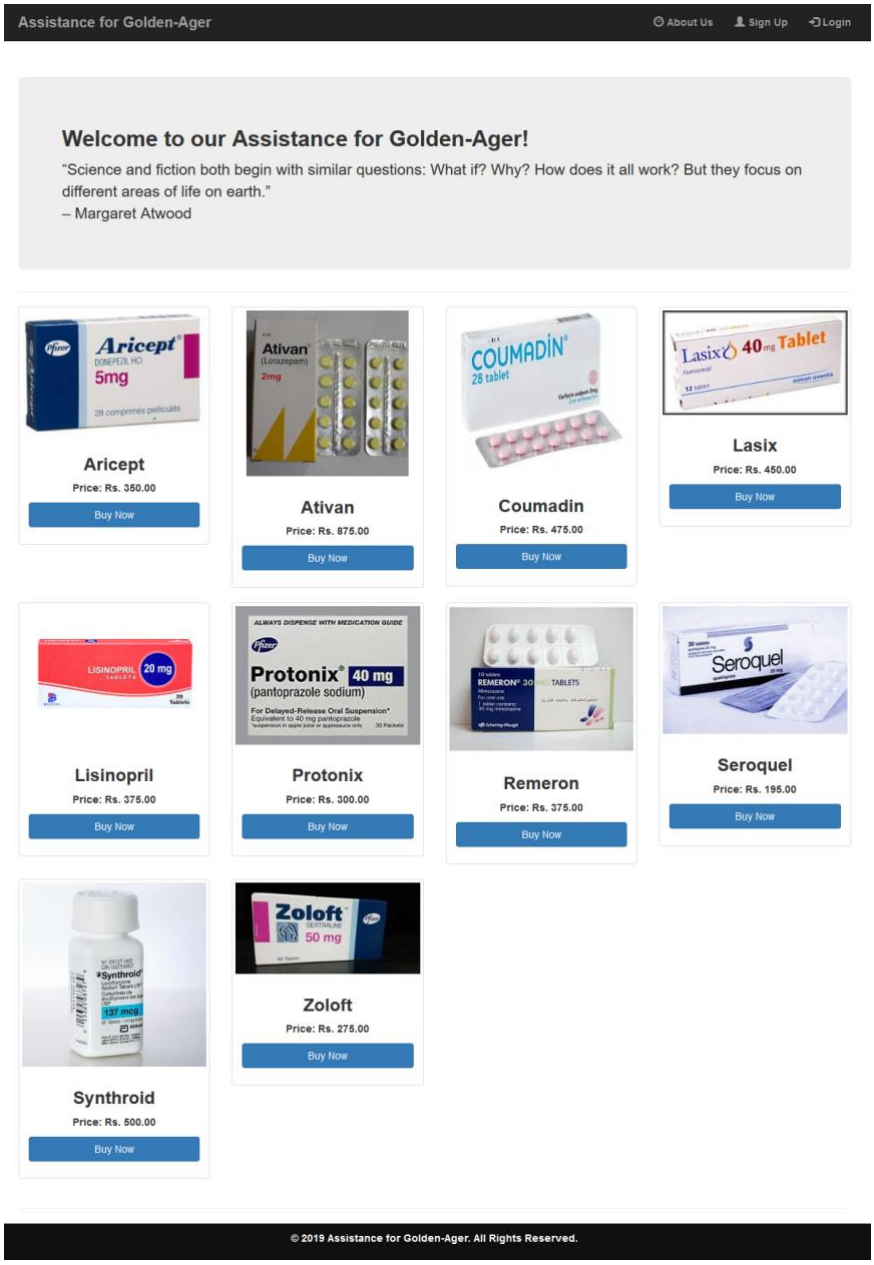

Fig 5.5: Purchase Medicines

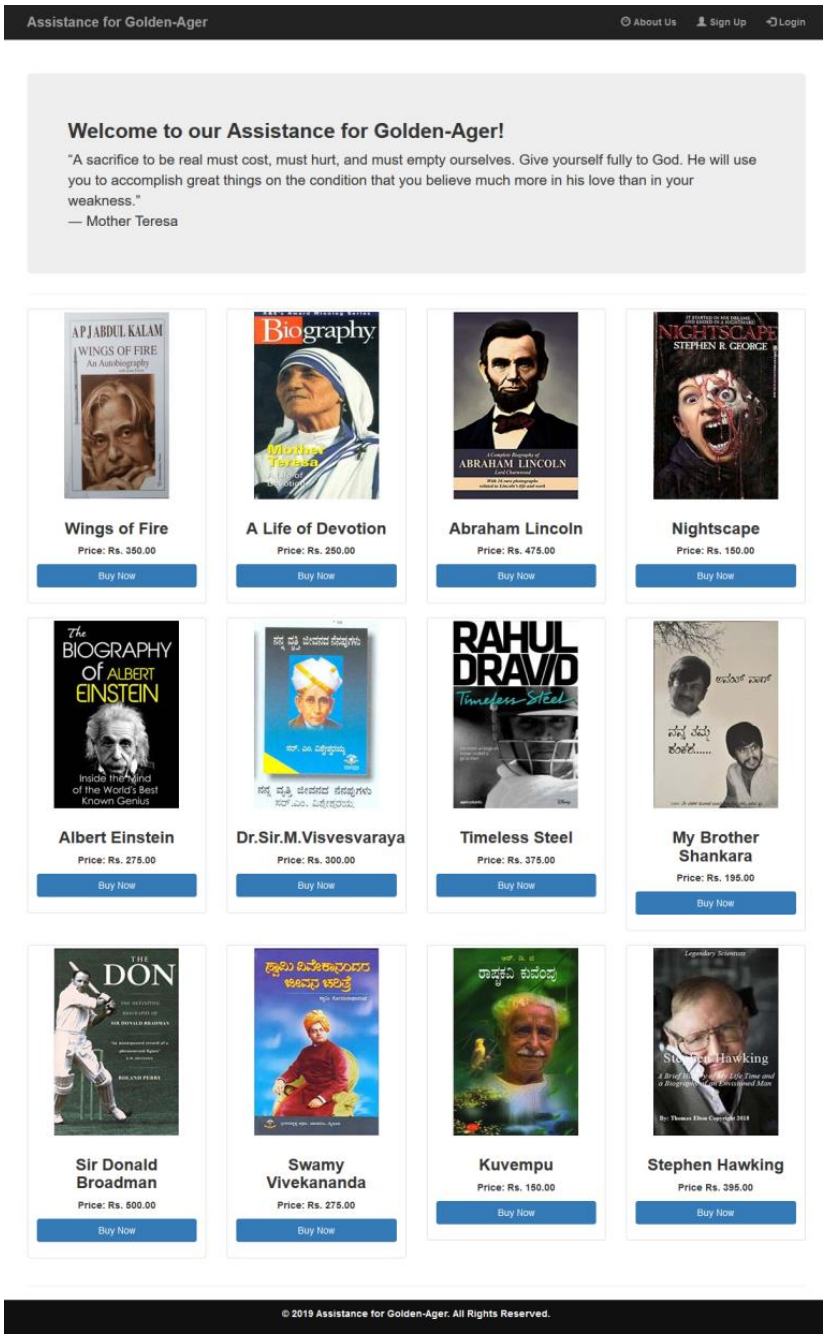

Fig 5.6: Purchase Bools

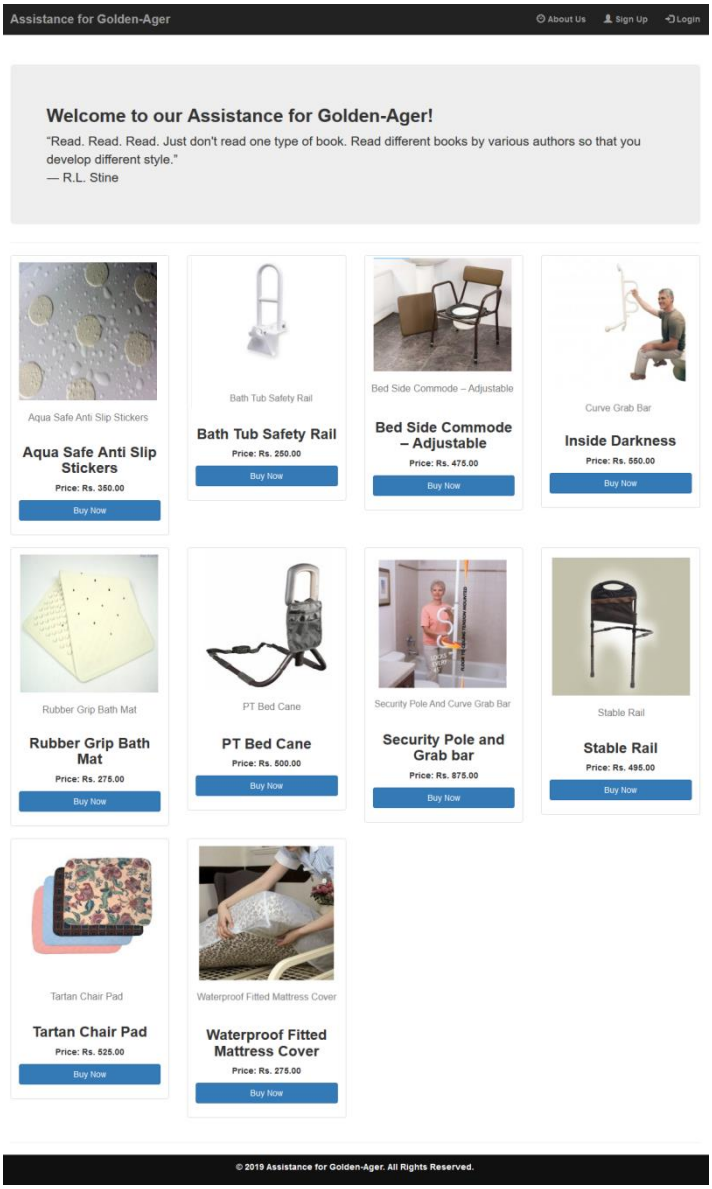

Fig 5.7: Purchase Support Aid.

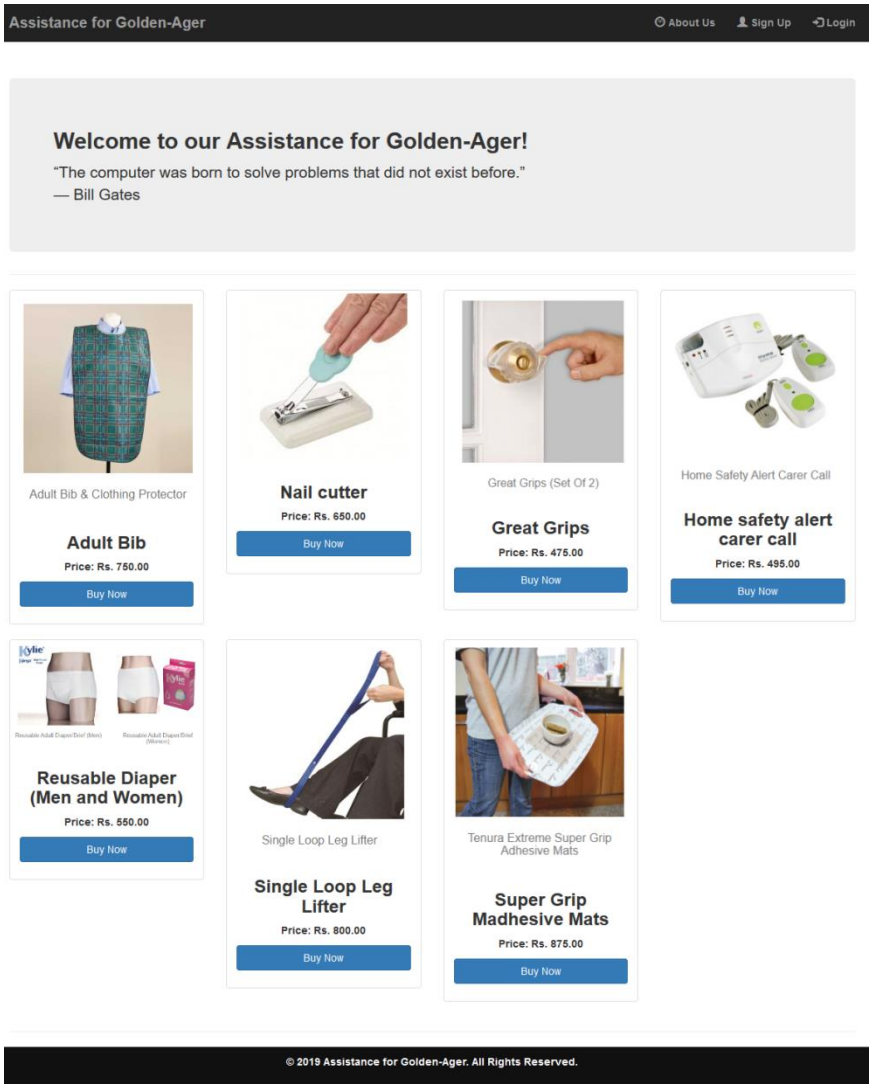

Figure 5.8: Purchase Daily Aid.

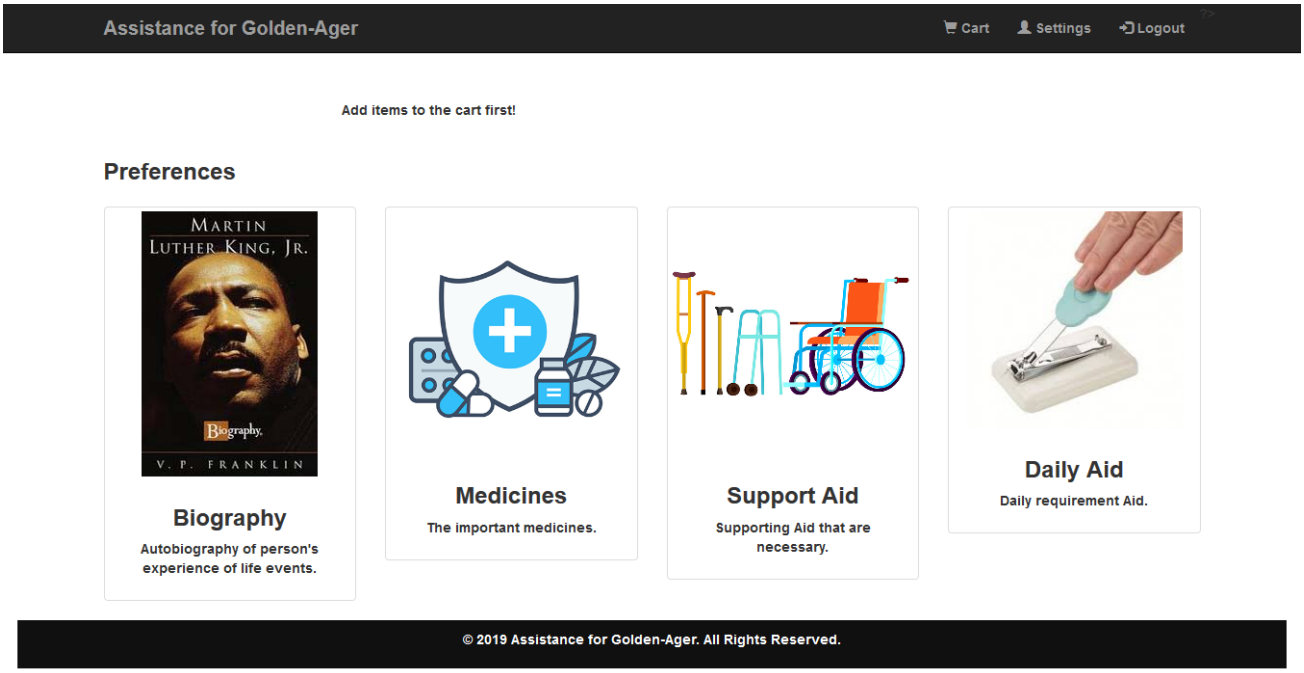

Figure 5.9: Empty Cart.

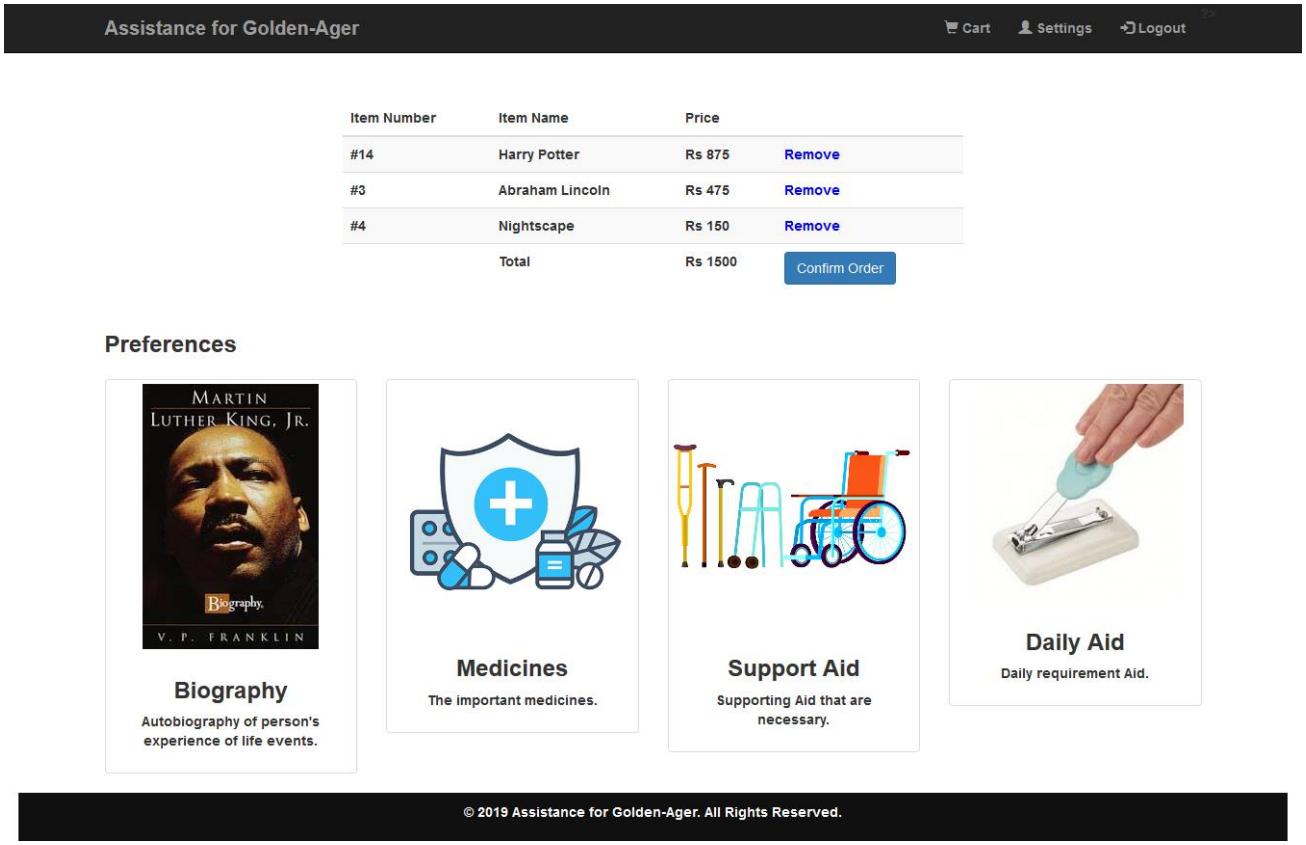

Figure 5.10: Cart.

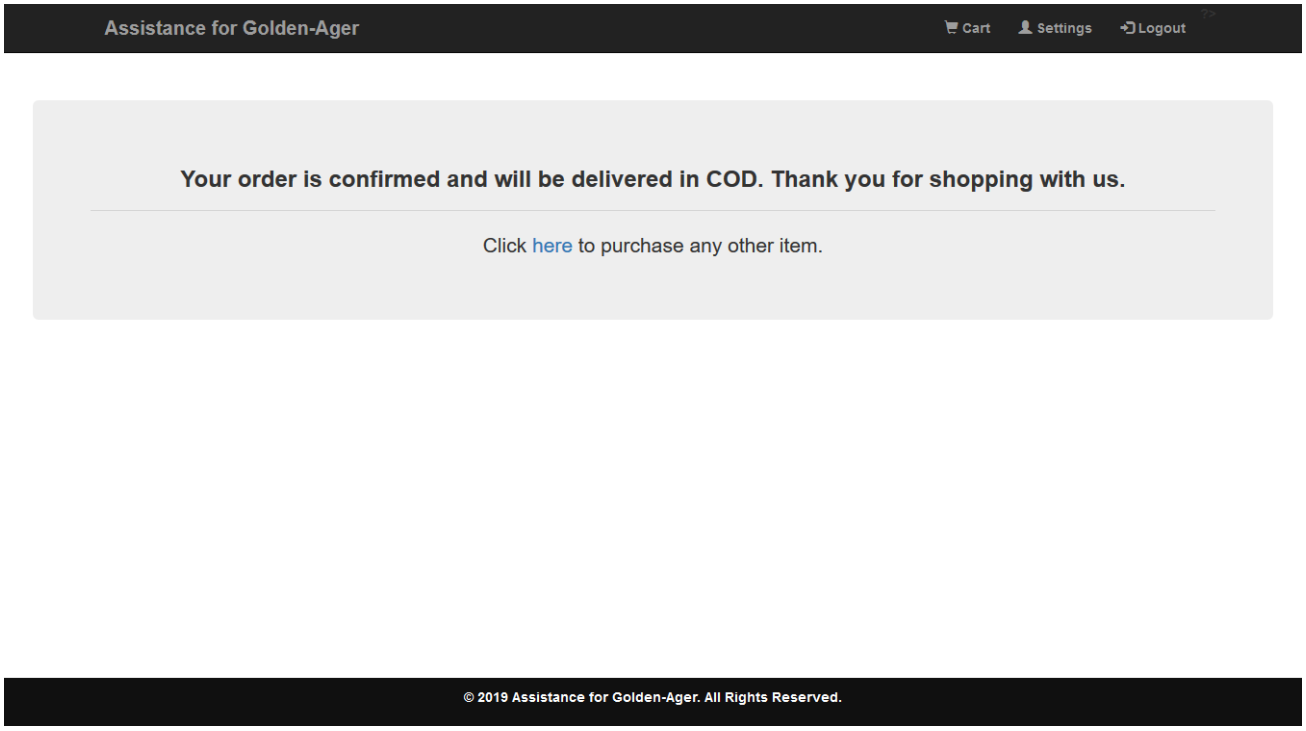

Figure 5.10: Shipping.

CONCLUSION

This system is a humble venture to satisfy the needs of the Elderlies of the society by providing an online service for their care, support and shopping. It enables the day-to-day sales functions with an idea to enable customers to order products and services using our web project. The system enables vendors to set up their products online, customers to browse through the shops, and a system administrator to approve and reject requests for new shops and maintain lists of shop categories. Several user friendly coding techniques have been used. Our agenda is to design an online shopping site that manages the products and simplifies customer's purchases without having to visit stores physically. Shopping will be highly personalized and the system aims to provide lower prices than most competitors in the market.

REFERENCES

[1]. https://www.w3schools.com/

[2]. https://www.seniorsafetyreviews.com/11-products-for-elderly-living-alone/

[3]. https://cssreference.io/

[4]. Robin Nixon "Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5", 4th Edition.

[5]. Luke Welling, Laura Thomson "PHP and MySQL Web Development ", 5th Edition.

[6]. Zak Ruvalcaba Anne Boehm "Murach's HTML5 and CSS" 3rd Edition

[7]. "Step by step Bootstrap 3" by Riwanto Megosinarso