CHAPTER 1

INTRODUCTION

1.1 Introduction to DBMS

Database is a collection of related data. DBMS came into existence in 1960 by Charles. Again in 1960 IBM brought IMS-Information Management System. In 1970 Edgar Codd at IBM came up with a new database called RDBMS. In 1980 came SQL Architecture-Structured Query Language. Between 1980 to 1990 there was advances in DBMS e.g. DB2, ORACLE. A database has the following implicit properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discovers (UOD). Changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot be referred to as a database.
- A database is designed, built and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents.

Metadata (metadata, or sometimes meta information) is “data about data”, of any sort in any media. An item of metadata may describe a collection of data including multiple content items and hierarchical levels, for example a database schema. In data processing, metadata is definitional data that provides information about or documentation of order data managed within an application or environment. The term should be used with caution as all data is about something, and is therefore metadata.

A database management system (DBMS) is a collection of programs that enables users to create and maintain the database. The DBMS is a general purpose software system that

facilitates the process of defining, constructing, manipulating and sharing database among various users and applications. Defining a database specifying the database involves specifying the data types, constraints and structures of the data to be stored in the database. The descriptive information is also stored in the database in the form of a database catalogue or dictionary; it is called meta-data. Manipulating the data includes querying the database to retrieve the specific data. An application program accesses the database by sending the queries or requesting for data for DBMS. The important function provided by the DBMS includes protecting the database and maintaining the database.

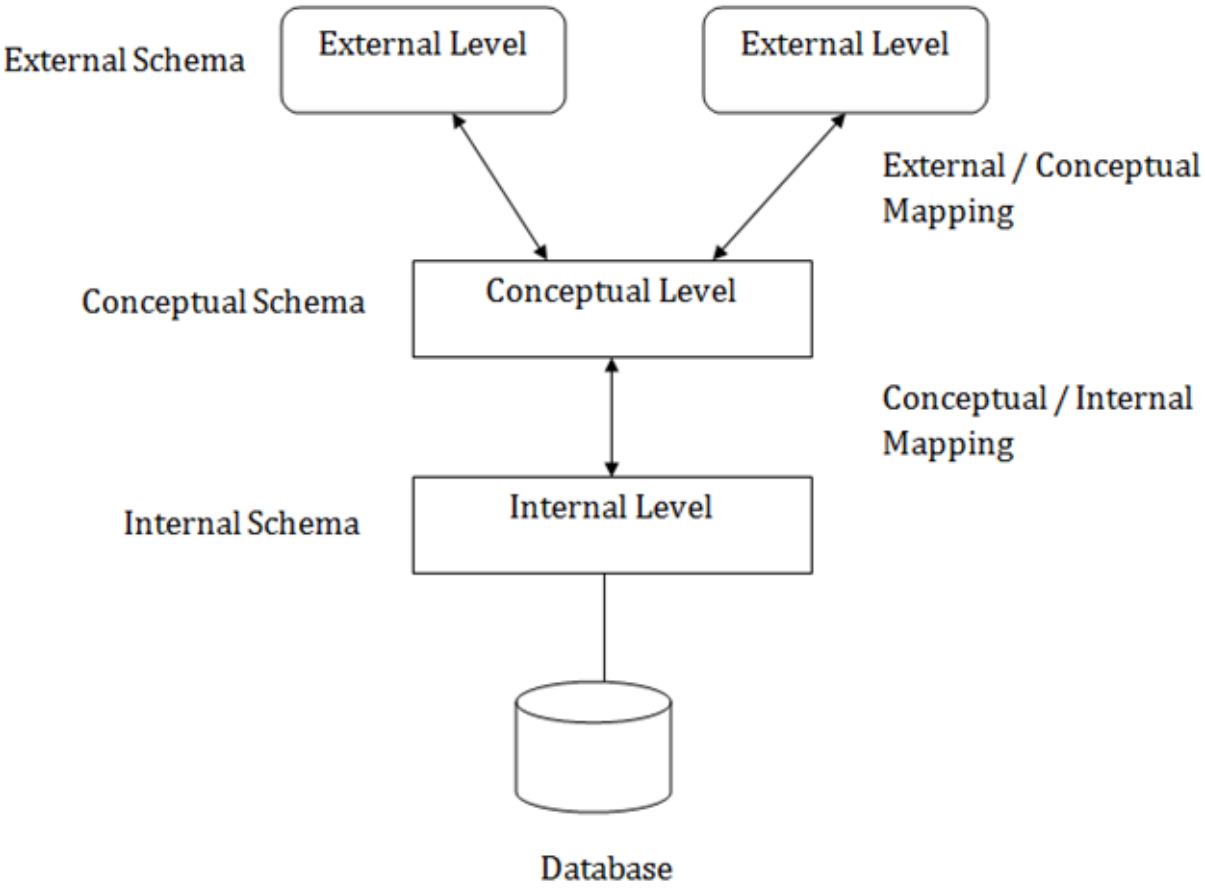

Figure 1.1: Three Schema Architecture

The Figure 1.1 shows the Three Schema Architecture of Database Management System. The architecture has three levels:

- **External level:**

  The external level is the view that the individual user of the database has. This view is often a restricted view of the database and the same database may provide a number of different views for different classes of users. In general, the end users and even the application programmers are only interested in a subset of the database.

- **Conceptual level:**

  The conceptual view is the informational model of the enterprise and contains the view of the whole enterprise without any concern for the physical implementation. The conceptual view is the overall community view of the database and it includes all the information that is going to be represented in the database.

- **Internal level:**

  The internal view is the view about the actual physical storage of data. It describes what data is stored in the database and how.

1.2 Problem statement

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser. Consumers can find product of interest by visiting these online shopping websites.

1.3 Objectives of the project

- The main objective of this project is to develop an online shopping system where a customer is provided with a shopping cart application using which they will be able to easily purchase products online from the comfort of home through the Internet by E-Commerce based application.
- The aim of this project is to develop a database containing data of products, users, purchase history which include insertion, deletion, update, view and retrieval of data
- This application will have a user friendly interface.
- Developing a secured database is also a main criterion.

- To store up-to-date information about customer's orders.
- The project is totally built at administrative end and thus only the administrator is granted permission to update or delete products, users and orders.
- The overall project design objective is to provide an efficient modular design that will reduce the system complexity, facilitate changes and result in an easy implementation.

CHAPTER 2

SYSTEM DESIGN AND METHODOLOGY

2.1 System Architecture

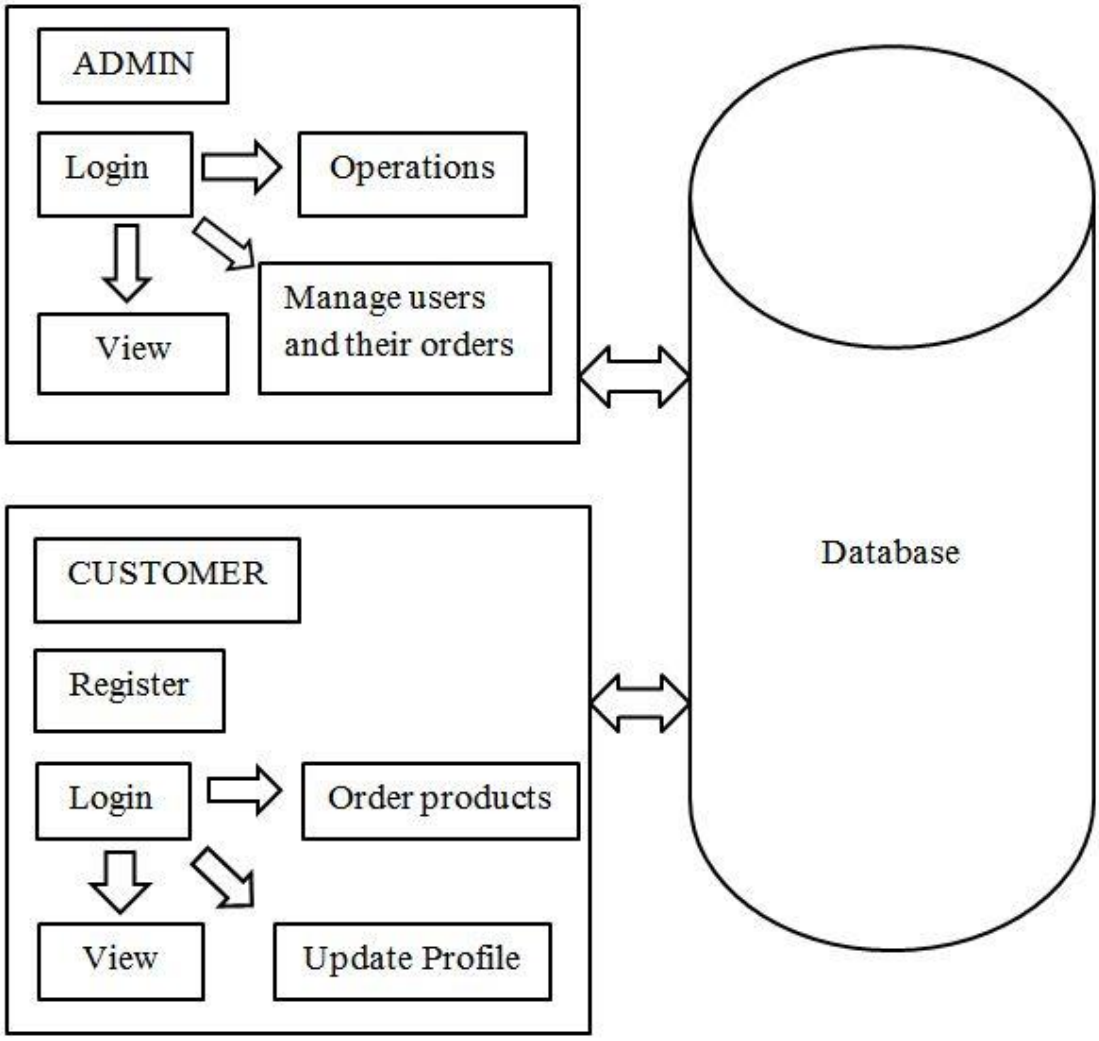

Figure 2.1: System Architecture of Online Shopping System

The above figure 2.1 describes the system architecture of Online Shopping System. The architecture consists of a centralized database, which can be accessed by admin and normal users or customers. Administrative access is required for the admin, which is implemented

through login module by which the admin can login with the registered username and password.

Once login is successful, the admin can manipulate the updating or deleting customers or users and their orders, adding, updating and deleting categories and products. If the admin fails to login, the admin gets a popup message that the username and password are incorrect .The admin needs to enter valid username and password once again.

A customer first needs to register in the website by filling up valid username, email and password. Username should be unique and if it is already been taken then a popup message is displayed that the username is already taken. Password confirmation should be made by retyping the password again. If the password and password confirmation do not match then it produces a popup message stating that they don't match each other.

On successful registration, customer can login through valid username and password. On successful login, customer is directed to products list where he can view products of his interest and then add to cart, select delivery address, make payment and order the products. Customer can also update his username, email and profile picture. He can also see his previous orders list.

2.2 Hardware Requirements

Processor	: Intel Core
Processor speed	: 13 GHz
Ram size	: 1 GB RAM (min)
Minimum disk space	: 250 GB space (min)

2.3 Software Requirements

Database	: SQLite
Front End	: HTML, BOOTSTRAP
Back End	: Python, Django

2.4 Development Environment

- **HTML**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

- **CSS (Cascading Style Sheets)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

- **Bootstrap**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden.

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

- **SQLite**

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program.

SQLite is ACID-compliant and implements most of the SQL standard, generally following PostgreSQL syntax. However, SQLite uses a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. This means that one can, for example, insert a string into a column defined as an integer. SQLite will attempt to convert data between formats where appropriate, the string "123" into an integer in this case, but does not guarantee such conversions, and will store the data as-is if such a conversion is not possible.

SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

- **Python**

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, is "sunsetting" on January 1, 2020 (after extension; first planned for 2015), and the Python team of volunteers will not fix security issues, or improve it in other ways after that date. With the end-of-life, only Python 3.5.x and later will be supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source

reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

- **Django**

Django is a Python-based free and open-source web framework, which follows the model-template-view (MTV) architectural pattern .It is maintained by the Django Software Foundation (DSF), an independent organization established as a 501(c)(3) non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Django was created in the fall of 2003, when the web programmers at the Lawrence Journal-World newspaper, Adrian Holovaty and Simon Willison, began using Python to build applications. Jacob Kaplan-Moss was hired early in Django's development shortly before Simon Willison's internship ended. It was released publicly under a BSD license in July 2005. The framework was named after guitarist Django Reinhardt.

2.5 Schema Diagram

A database schema is a skeleton structure that represents the logical view of the entire database. It defines tables, views and integrity constraints.

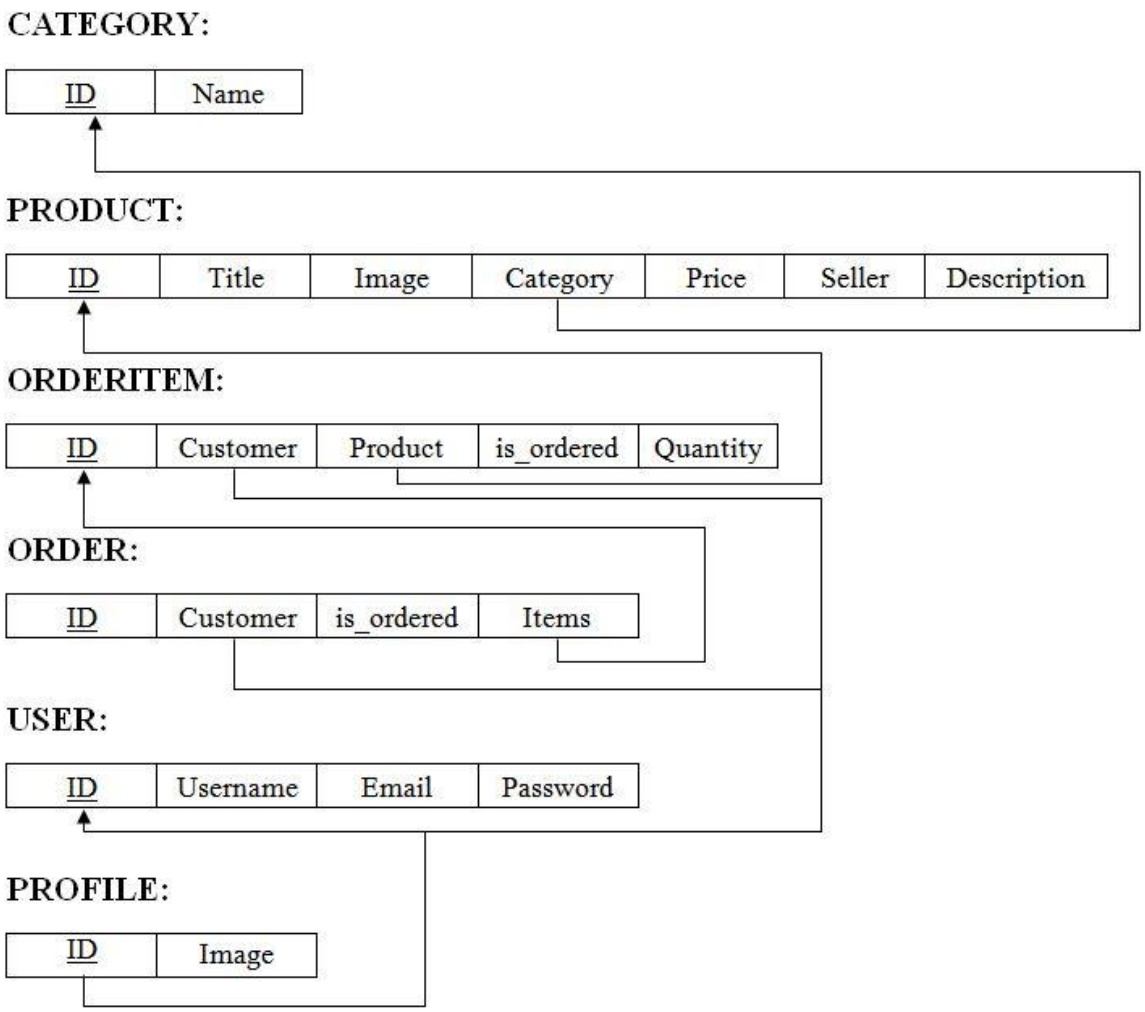

Figure 2.5: Schema Diagram

2.6 ER Diagram

An entity-relationship model describes inter-related things of interest in specific domain of knowledge. An ER module is composed of entity types and specifies relationships that can exist between instances of those entity types. It is a data modeling technique that graphically illustrates an information systems entities and the relationship between those entities.

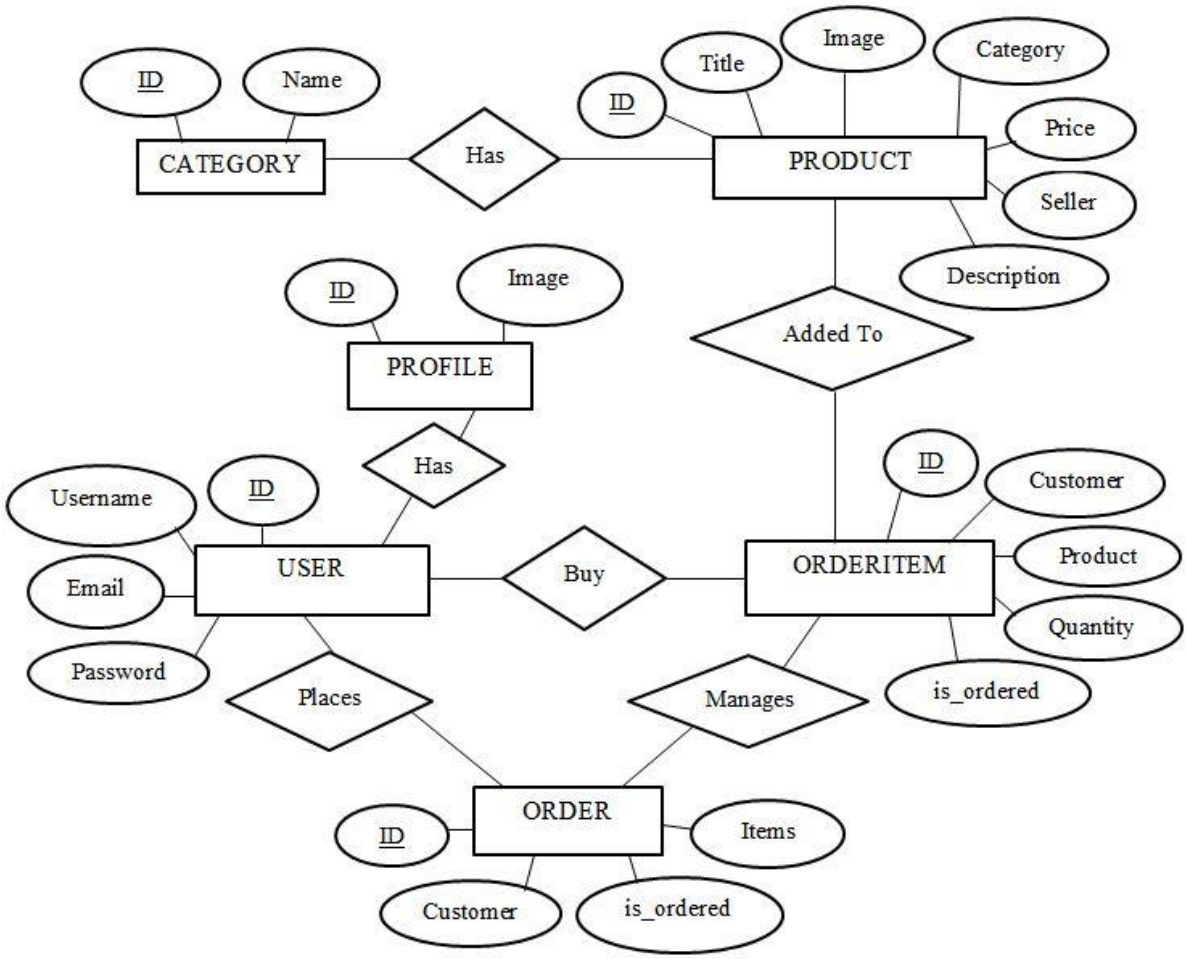

Figure 2.6: ER Diagram

CHAPTER 3

SYSTEM IMPLEMENTATION

To implement this system, Django, SQLite, Python is used for backend and HTML, Bootstrap, CSS, Python is used for frontend.

3.1 Admin module:

INPUT:

The login inputs are Username and password.

OUTPUT:

A successful login shall direct admin to product list page. If the login attempt fails, popup message will be displayed that the username and password are invalid and admin needs to retype the username and password again.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. Login page consists of a form with username, password fields. On filling the form the admin should click on the submit button. If the username and passwords are valid then admin is directed to Product list page. If not using messages framework of Django, a danger message will be displayed stating that the username and password are invalid.

OBJECTIVES:

- The main purpose of this module is to provide all the functionality such as create, update and delete options to categories, products, orders, orderitems, users and their profiles only to admin and not for customers.
- Admin can view list of users and only he can update user's details or can also delete users.
- Admin can view list of categories and products

- Only admin can create new categories and products, update or delete existing categories and products.
- Admin can manipulate the orders and orderitems of customers (users).

3.2 Customer module:

INPUT:

Inputs for registration are Username, Email, Password, Password confirmation. Inputs for login are Username, Password.

OUTPUT:

On successful registration, customer will be redirected to login page to login. If not popup message will displayed informing customer that the username is already taken or invalid email or password did not match with password confirmation.

A successful login shall direct customer to product list page. If the login attempt fails, popup message will be displayed that the username and password are invalid and customer needs to retype the username and password again.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. Registration page consists of a form with username, email, password fields. Login page consists of a form with username, password fields. On filling the form the customer should click on the submit button. If the username and passwords are valid then admin is directed to Product list page. If not using messages framework of Django, a danger message will be displayed stating that the username and password are invalid.

OBJECTIVES:

- The main objective of customer module is to provide all the required limited functionality to customers.
- Customer can view the list and details of categories and products.

- Customer can add products to his cart, change the quantity of products required, select delivery address, make payment and then order the products.
- Customer can view his profile and can also update his username, email and profile picture. He can also view his previous orders list.

3.3 Product module:

INPUT:

Inputs for product module are title, product image, category, price, seller, description.

OUTPUT:

The product will be created, updated or deleted based on admin input.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. Only admin can manage product module. If admin wants to create or update a new product he can click the respective buttons. He then needs to fill all the details regarding the product and then submit which will create or update products. Customer can view this product but not modify it. Admin can also delete products.

OBJECTIVES:

- The main purpose for developing this module is to manage the products.
- All products will be managed by admin and customer will be able to see product.
- Admin can create a new product or update an existing product.
- Admin can even delete a product.

3.4 Category module:

INPUT:

Input for category module is name.

OUTPUT:

A new category will be created or existing category will be updated or deleted.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. Only admin can manage category module. If admin wants to create or update a new category he needs to click the respective buttons. He then needs to fill the category name and then submit which will create or update category. Customer can view this category but not modify it. Admin can also delete category.

OBJECTIVES:

- The main purpose for developing this module is to manage the categories.
- All categories will be managed by admin and customer will be able to see category.
- Admin can create a new category or update an existing category.
- Admin can even delete a category.

3.5 Order module

INPUT:

Inputs for this module are customer, is_ordered, items.

OUTPUT:

An order will be created when customer adds products to cart or when he orders.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. When customer adds or removes products from cart, order will be created for that particular customer. If the products are not yet ordered, is_ordered will be set to False. If it is ordered is_ordered will be set to True. Order module also has a method to calculate total price.

```
def get_cart_total(self):
    total=0
    for order_item in self.items.all():
        total+=order_item.totalprice()
return total
```

OBJECTIVES:

- The main objective of this module is to manage customer's orders.
- Only admin can manage order module.
- Customer can order items and can view his previous orders.
- Admin can view orders of all customers and can manipulate them.

3.6 Orderitem module:

INPUT:

Inputs for orderitem module are customer, product, is_ordered, quantity.

OUTPUT:

An orderitem will be created which will be added to items of order module.The quantity of product can be specified.

DESCRIPTION:

Frontend is designed using HTML, Bootstrap. Customer when adds product to cart, an orderitem will be created which acts as a link between product module and order module. If the product is ordered, is_ordered is set to True. If not, is_ordered is set to False. Orderitem module also has a function to calculate total price of the product which will be used by order module.

```
def totalprice(self):
    return self.product.price*self.quantity
```

OBJECTIVES:

- The main objective of this module is to manage customer's orders.
- This module acts a bridge between product module and order module.
- Only admin can manage orderitem module.
- Customer can change the quantity of products required which will be stored in this module.
- Admin can view orderitems of all customers and can manipulate them.

CHAPTER 4

RESULTS AND SCREENSHOTS

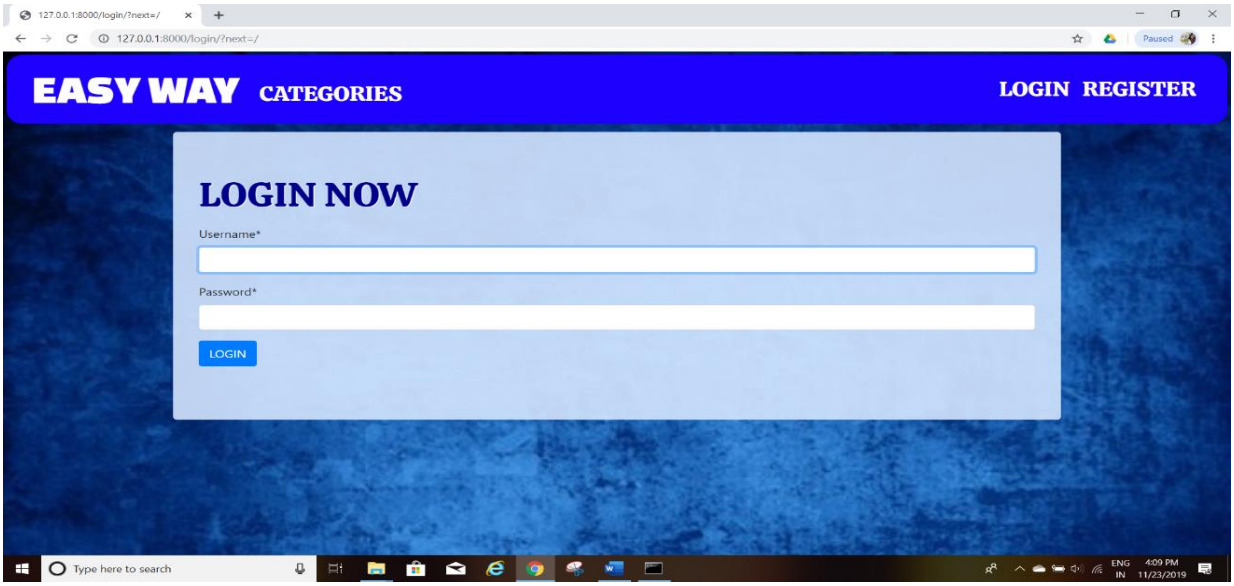

Figure 4.1: Login page

Figure 4.1 represents a login page through which customer or admin can login.

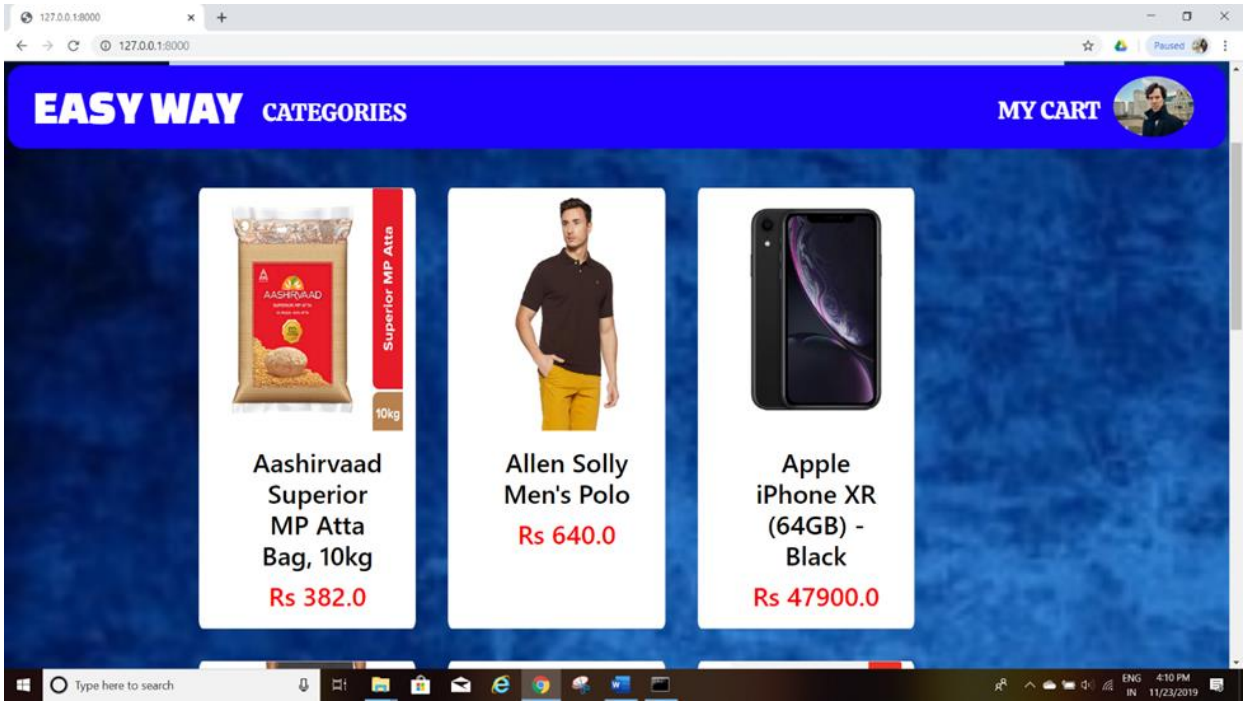

Figure 4.2: Product list page

Figure 4.2 represents product list page in which all the products are displayed for both customers and admin.

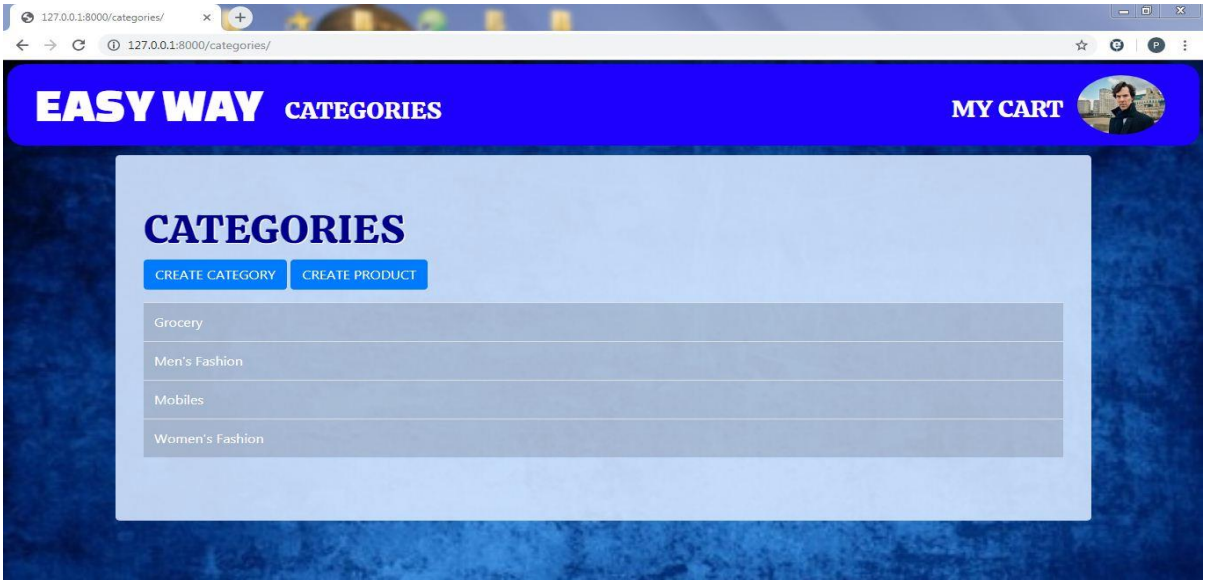

Figure 4.3: Categories Admin page

Figure 4.3 represents admin categories page.

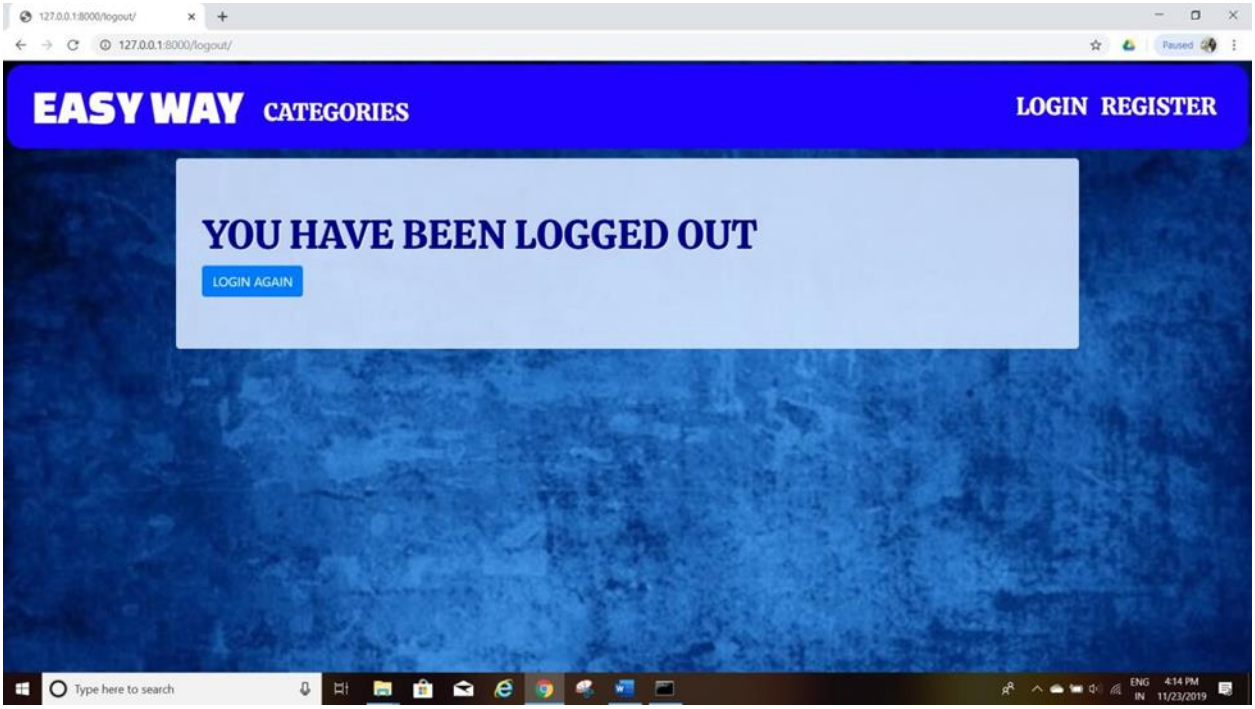

Figure 4.4: Logout page

Figure 4.4 represents a logout page which is displayed when the customer or admin logs out.

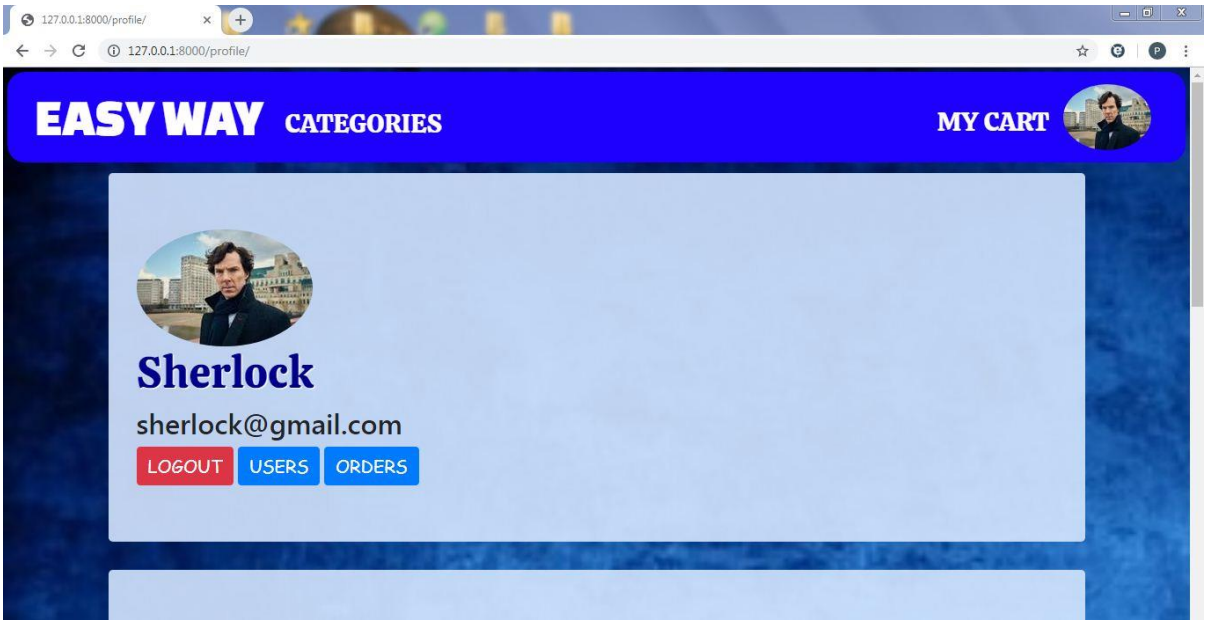

Figure 4.5: Admin profile

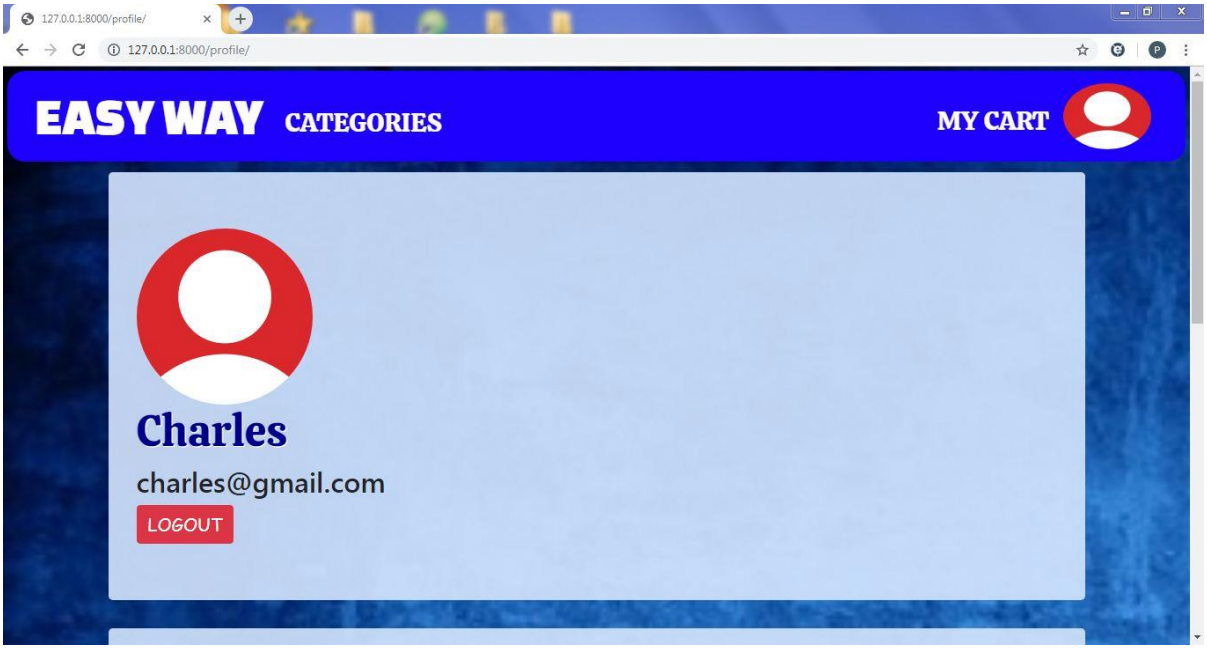

Figure 4.6: Customer profile

Figure 4.5 and 4.6 represents Admin profile and customer profile which gives details about the customer and admin. It even has logout button to logout. Admin has a users button to view all users and orders button to view all orders.

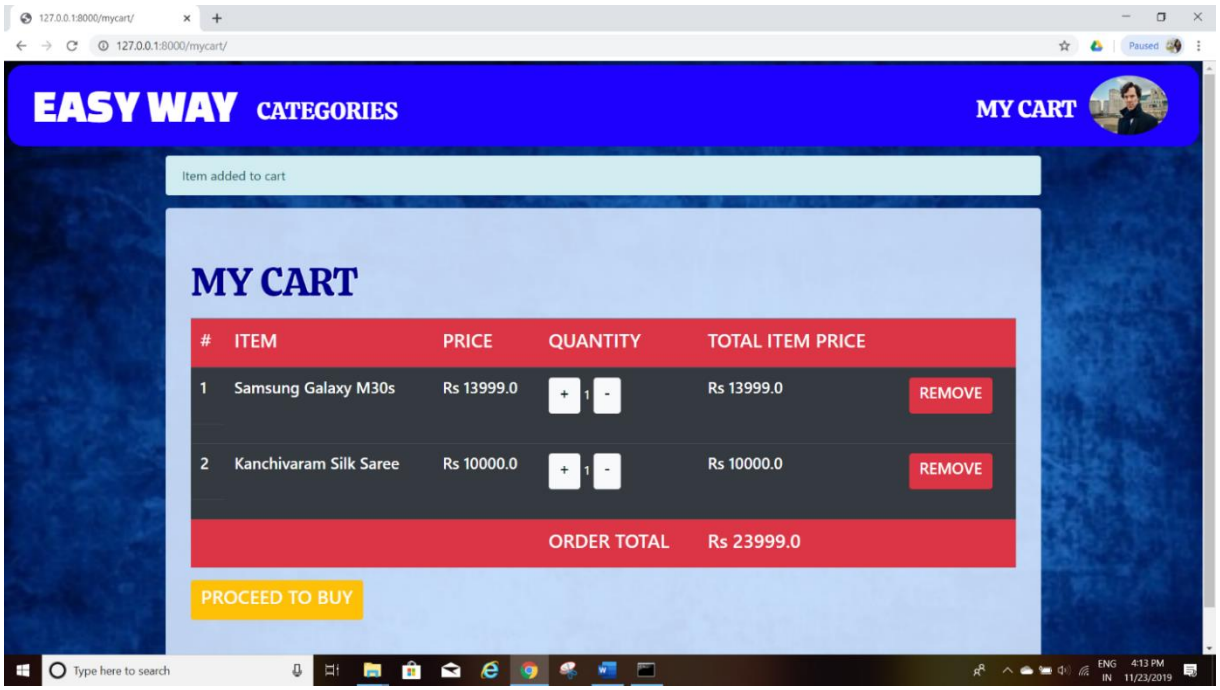

Figure 4.7: Customer's Cart

Figure 4.7 represents customer's cart which has list of products which customer wants to buy.

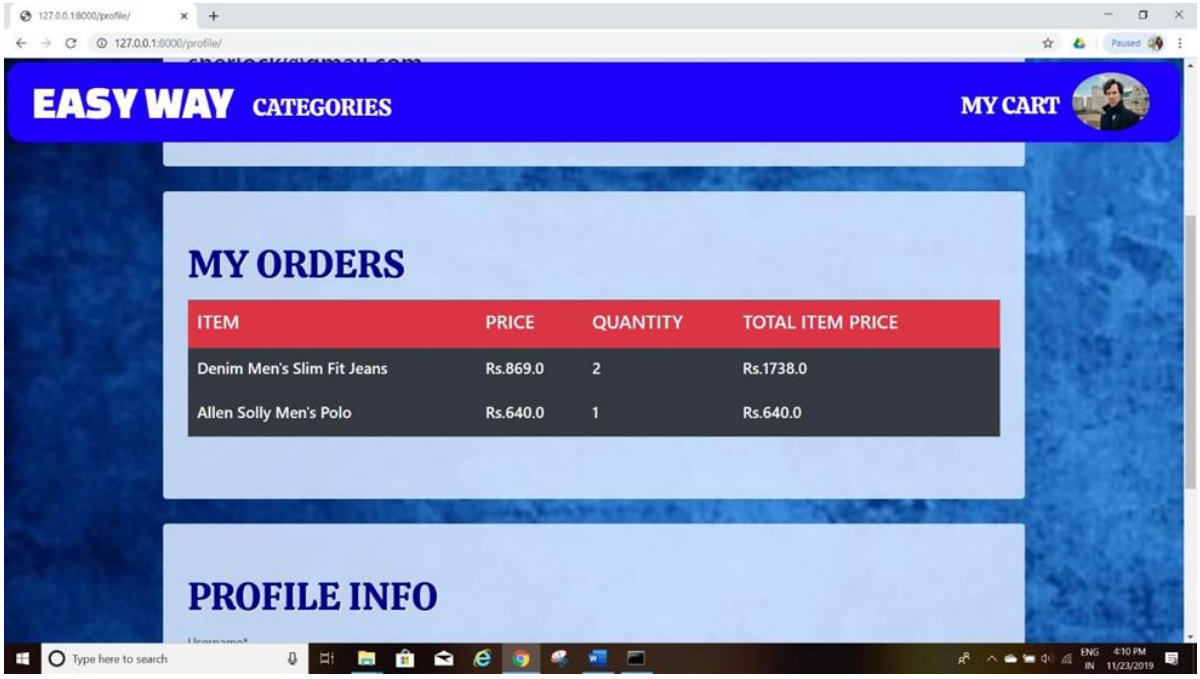

Figure 4.8: Customer's orders history

Figure 4.8 represents customer's previous orders history which is visible in his profile page.

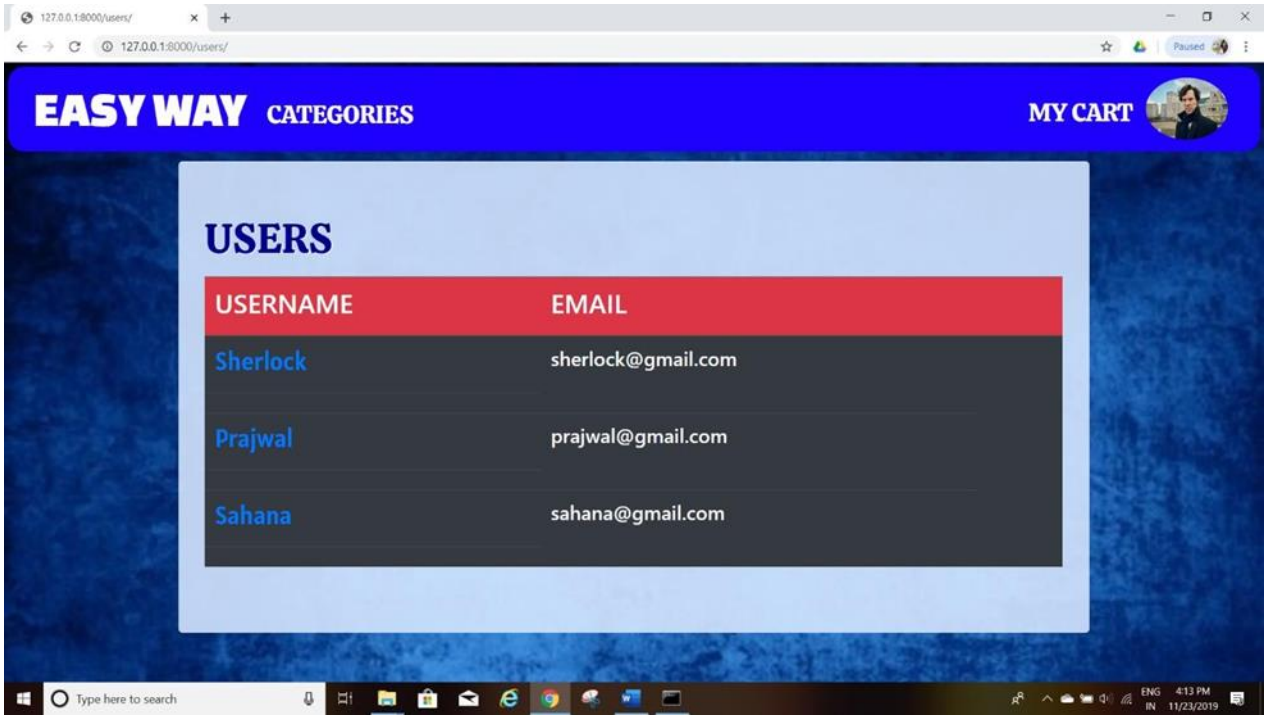

Figure 4.9: Users list

Figure 4.9 represents a list of users which can only be viewed and edited by admin.

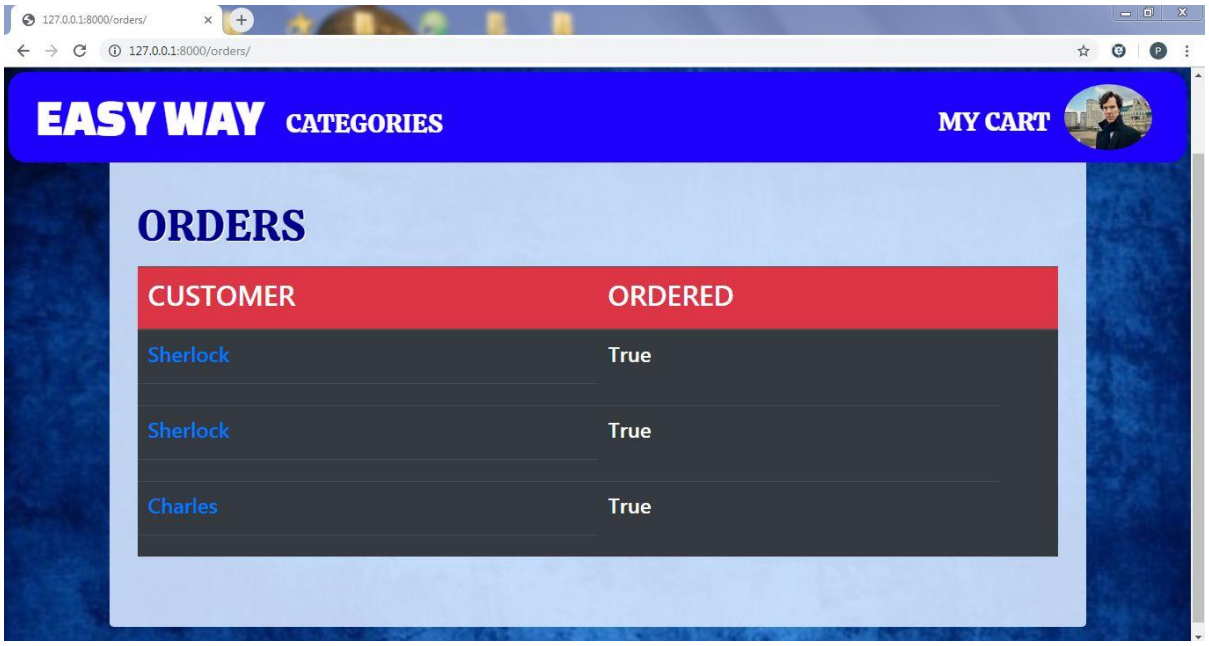

Figure 4.10: Orders list

Figure 4.10 represents a list of orders of all customers which can only be viewed and edited by admin.

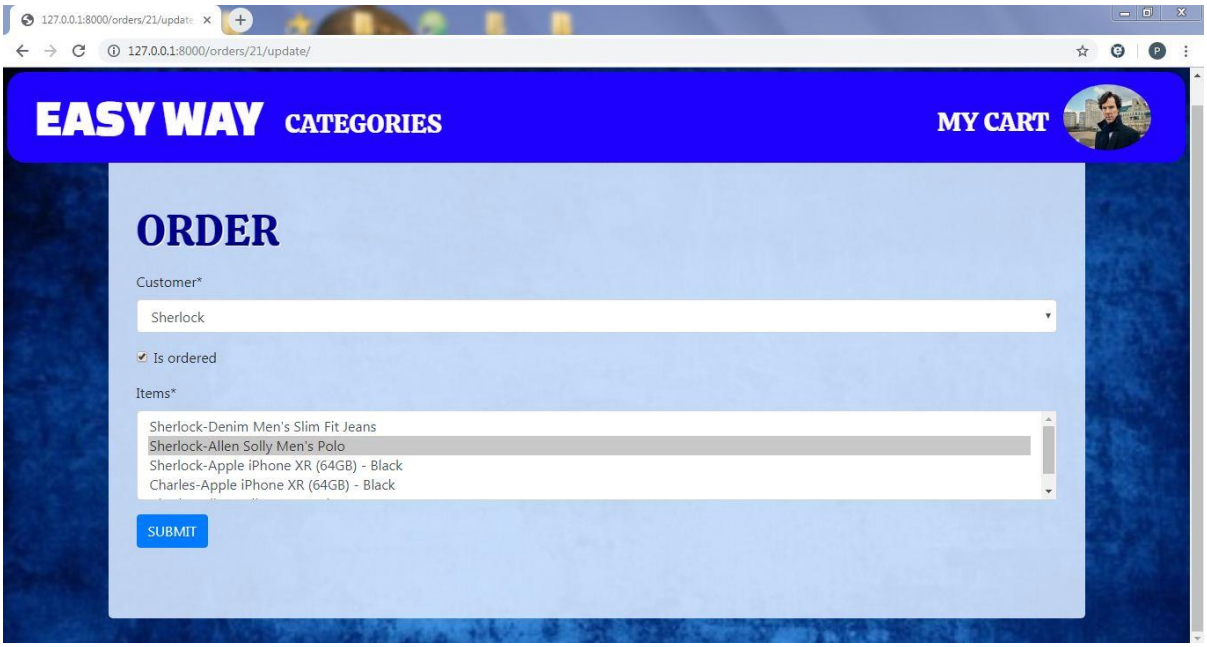

Figure 4.11: Order update

Figure 4.11 represents order update which can only be done by admin.

Figure 4.12: Orderitem update

Figure 4.12 represents orderitem update which can only be done by admin.

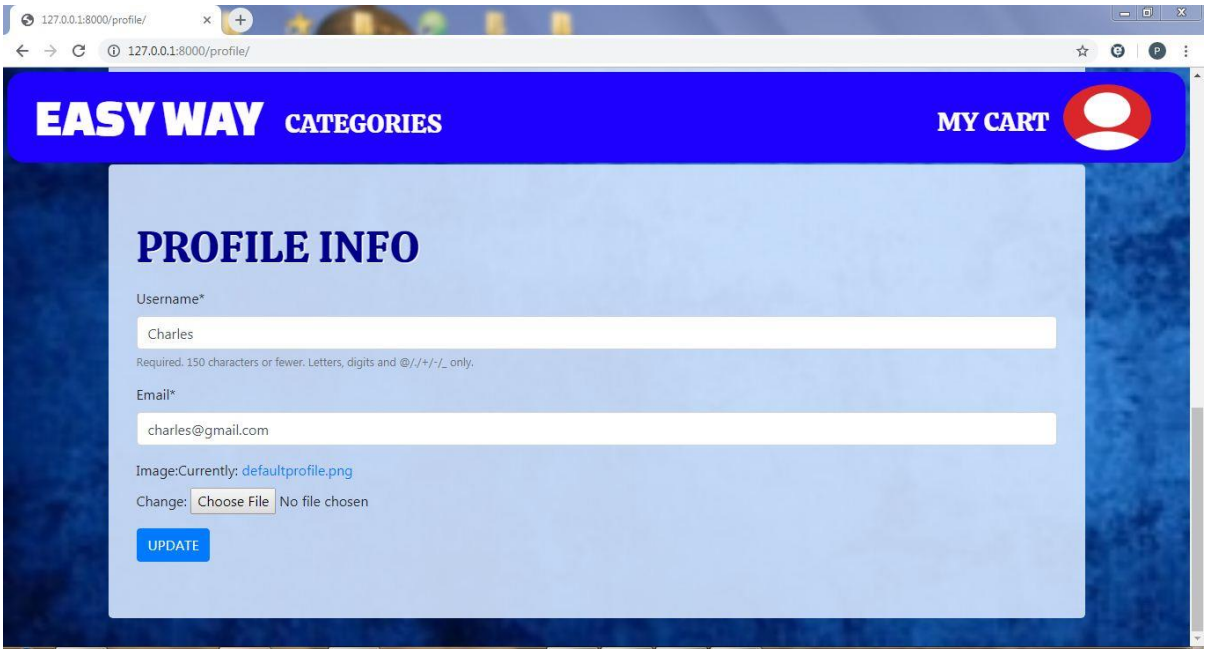

Figure 4.13: Customer Profile Update

Figure 4.13 represents customer profile update which can be done both by admin and customer.

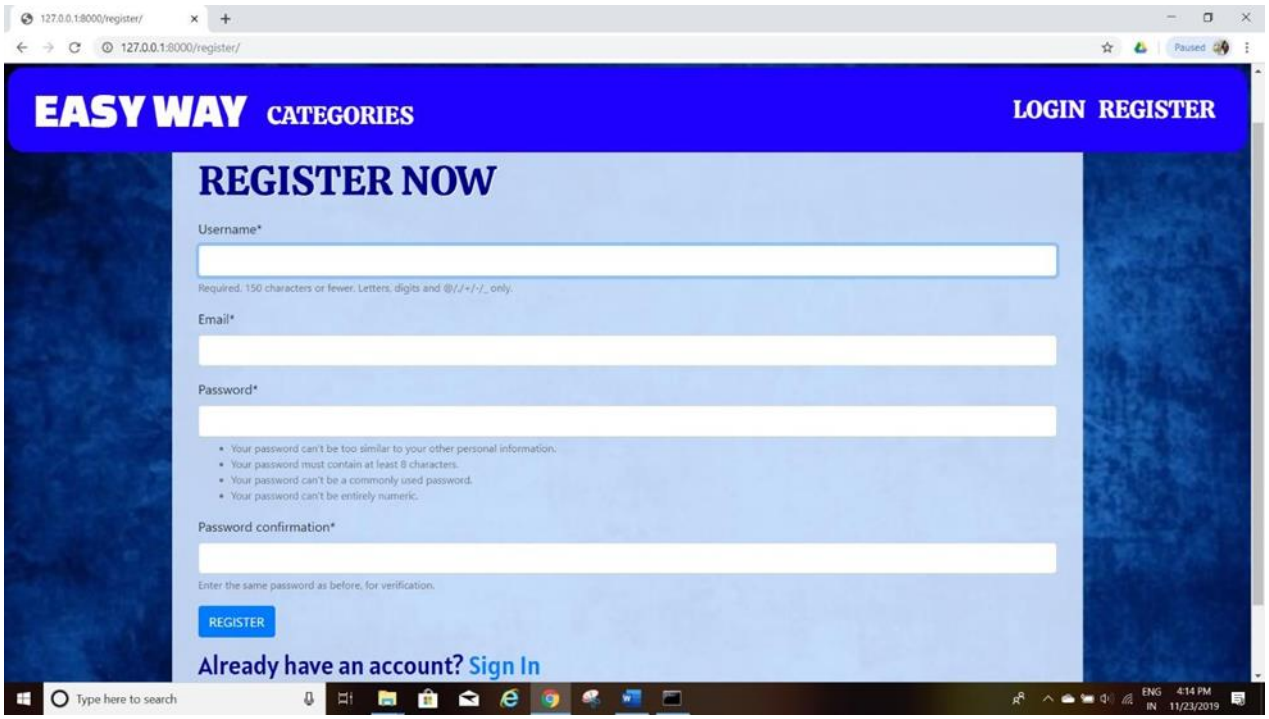

Figure 4.14: Register page

Figure 4.14 represents a registration page which is used to register into the online shopping portal

CHAPTER 5

CONCLUSION AND FUTURE WORKS

"Online Shopping System" is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser. Consumers can find product of interest by visiting these online shopping websites. It is designed to reduce time for customers as it provides wide variety of goods and services which the customer can buy.

FUTURE WORKS:

- Customer's activity can be studied and offers based on their activity for specific products can be given.
- Tailor-made product list can be created specific to a customer.