VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JNANA SANGAMA", BELAGAVI-590018

A DBMS LABORATORY AND MINI PROJECT (18CSL58)

REPORT ON

E-COMMERCE

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Engineering

In

Artificial Intelligence & Machine Learning

By

LIKHITH V [1KS20AI017]

Under the guidance of

Ms. Lakshmi K K
Asst. Prof, Dept. Of AIML

Ms. Anu Mathews
Asst. Prof, Dept. Of AIML

Department of Artificial Intelligence & Machine Learning

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

This is to certify that Mini Project work entitled **"E-COMMERCE"** is carried out by **Mr LIKHITH.V** bearing USN **1KS20AI017** bonafide student of **K.S. Institute of Technology** in the partial fulfilment for the award of the **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University, Belagavi,** during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said degree for the Fifth semester.

Dr. Vaneeta M
Associate Prof & HOD, AIML Department

Dr. Dilip Kumar K
Principal/Director, KSIT

Ms. Lakshmi K K
Asst. Prof, Dept. Of AIML

Ms. Anu Mathews
Asst. Prof, Dept. Of AIML

Name of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

I take this opportunity to thank everyone involved in the successful implementation of this mini project. I would like to thank the college for providing me an opportunity to work on the mini project.

I take this opportunity to express my sincere gratitude to my college **K.S. Institute of Technology,** Bengaluru for providing the environment to work on this mini project.

I would like to express my gratitude to our **MANAGEMENT,** K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the support provided for carrying out this mini project work in college.

I would like to express my gratitude to **Dr. K.V.A Balaji, CEO**, K.S. Institute of Technology, Bengaluru, for his valuable guidance.

I would like to express my gratitude to **Dr. Dilip Kumar K**, **Principal/Director**, K.S. Institute of Technology, Bengaluru, for his continuous support.

I like to extend my gratitude to **Dr. Vaneeta M**, **Associate Professor and Head**, Department of Artificial Intelligence & Machine Learning, for providing very good facilities and all the support provided in carrying out this Mini Project successfully.

I also like to thank my Mini Project Coordinators, **Ms. Anu Mathews, Asst. Professor, Ms. Lakshmi K K, Asst. Professor,** Department of Artificial Intelligence & Machine Learning for their help and support provided to carry out the Mini Project work successfully.

I am also thankful to the teaching and non-teaching staff of Artificial Intelligence & Machine Learning Department, KSIT for the help provided in completing this Mini Project.

DONE BY:

LIKHITH.V

USN: 1KS20AI017

ABSTRACT

The project report will provide an overview of the development process and design of the shopping website. Furthermore, it will discuss the impact of this website on customer satisfaction and convenience, as well as on overall profits. Finally, the website will have a secure purchase system, providing customers with an extra layer of protection when buying products. All of these features aim to create an efficient and safe online shopping experience for customers of Mart.

Chapter 1

INTRODUCTION

1.1 OVERVIEW

Online shopping is becoming increasingly popular as a way to find products, compare prices, and make purchases. With the convenience of the internet, consumers can quickly research companies, read reviews from other customers, and purchase products from the comfort of their own home. This report focuses on the development of an online shopping website for a project. The goal of this website is to provide a platform for consumers to easily and securely make purchases with the company(Mart). The custom website that is being developed is exclusively for Marts. This website will give customers the ability to quickly search for items, compare prices, and make secure purchases. The website will also have a user-friendly interface, making it easy for customers to navigate and find what they are looking for. Additionally, the website will have detailed descriptions of products , giving consumers an informed decision when making their purchase.

1.2 PROBLEM STATEMENT

The main aim of "E-COMMERCE WEBSITE" is to provide an easy interface for the Customers to view and buy products and help admin manage all the data.

1.3 DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. The DBMS essentially serves as an interface between the database and end users application programs, ensuring that data is consistently organized and remains easily accessible.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified ,and the database schema, which defines the database's logical structure. These three foundational elements help to provide concurrency, security, data integrity and uniform administration procedures. Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity.

1.4 SQL

SQL is a standard language for storing, manipulating and retrieving data in databases.

Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.[13]Since then, the standard has been revised to include a larger set of features. Despite the existence of such standards, most SQL code is not completely portable among different database systems without adjustments.

1.5 HTML / JavaScript

HTML is a markup language used for structuring and presenting content on the web and the fifth and current major version of the HTML standard.

HTML5 includes detailed processing models to encourage more interoperable implementations; it extends, improves and rationalizes the markup available for documents, and introduces markup and application programming interfaces (APIs) for complex web applications.

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

Chapter 2

REQUIREMENTS SPECIFICATION

A computerized way of handling information about property and users details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 OVERALL DESCRIPTION

A reliable and scalable database driven web application with security features that is easy to use and maintain is the requisite.

2.2 SPECIFIC REQUIREMENTS

The specific requirements of the E-commerce are stated as follows:

2.2.1 SOFTWARE REQUIREMENTS

Web Browser – Firefox 5.0 or later, Google Chrome – 6.0 or later , Brave
Database support - MySQL 5.7 or MySQL Server 5.7 or MySQL Shell 1.0.10 MySQL Workbench
Operating system – Windows 7 or above/ Ubuntu 16.04 or above

2.2.2 HARDWARE REQUIREMENTS

Processor – Pentium IV or above

RAM – 2 GB or more

Hard disk – 3 GB or more

Monitor – VGA of 1024x768 screen resolution

2.3 TECHNOLOGY

HTML is used for the front end design. It provides a means to structure text based information in a document. It allows users to produce web pages that include text, graphics and hyperlinks.
CSS (Cascading Style Sheets) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document.
SQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation.

Chapter 3

3. DETAILED DESIGN

3.1 SYSTEM DESIGN

The web server needs a JSP engine, i.e., a container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs. This server will act as a mediator between the client browser and a database.

The following diagram Fig . 3.1 shows the JSP architecture.

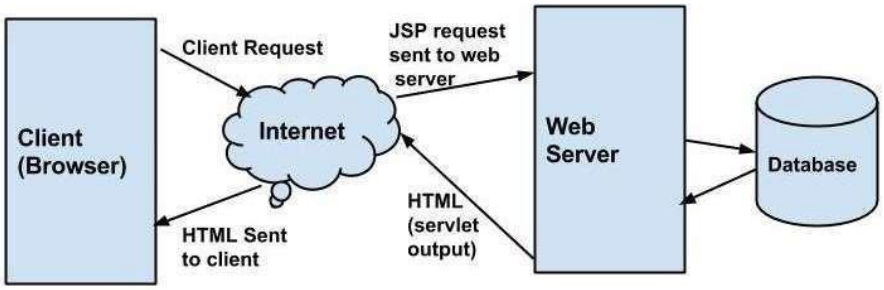

Fig. 3.1 JSP Architecture

Three-tier Client / Server database architecture is commonly used architecture for web applications. Intermediate layer called Application server or Web Server stores the web connectivity software and the business logic (constraints) part of application used to access the right amount of data from the database server. This layer acts like medium for sending partially processed data between the database server and the client. Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions. A database architect develops and implements software to meet the needs of users. Several types of databases, including relational or multimedia, may be created. Additionally, database

architects may use one of several languages to create databases, such as structured query language.

3.2 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the three schema approach to software engineering. While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ER Diagram is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Cardinality notations define the attributes of the relationship between the entities. Cardinalities can denote that an entity is optional. Fig .3.2.1 shows the ER diagram of E-commerce

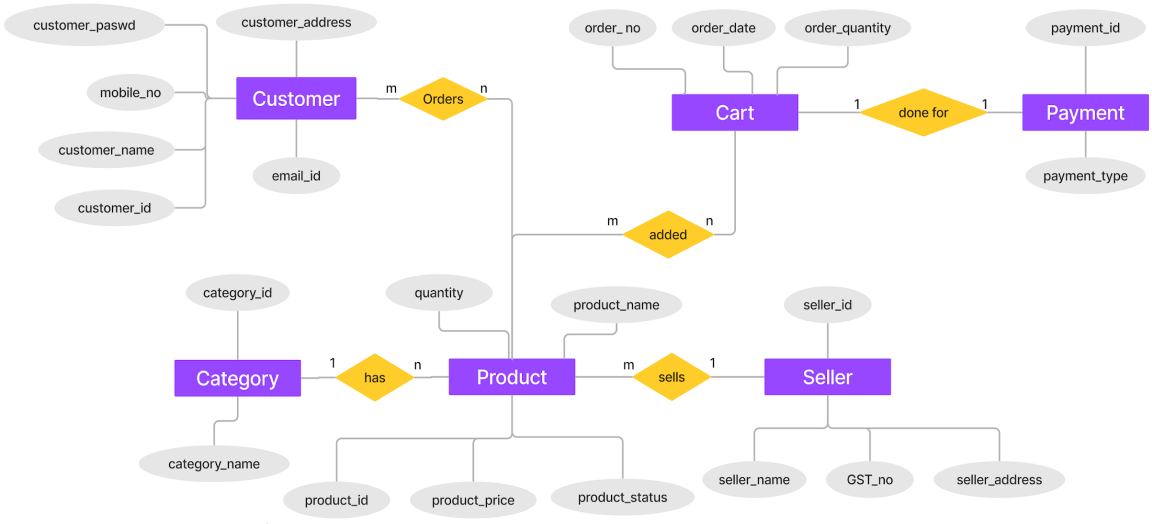

Fig.3.2.1 Enhanced ER diagram of E-Commerce

3.3 RELATIONAL SCHEMA

The term "schema" refers to the organization of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organize values in the database. The following diagram shows the schema diagram for the database.

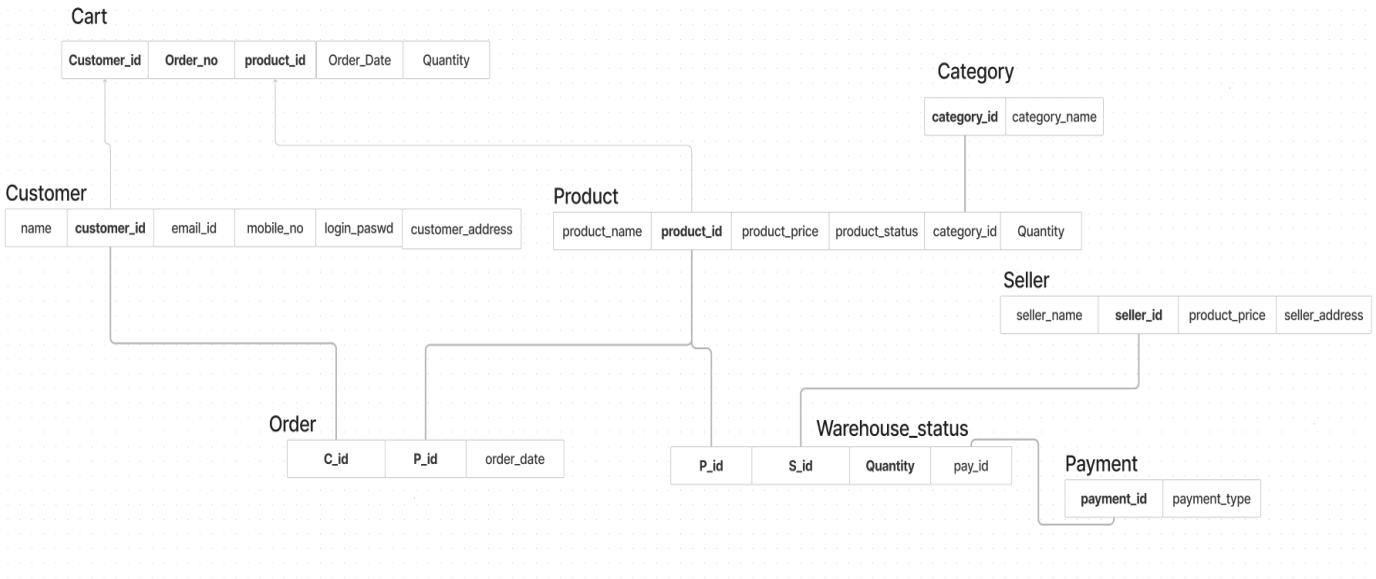

Fig. 3.3, Schema diagram

3.4 DESCRIPTION OF TABLES

The database consists of six tables:

1. Customer: It stores the customers details.

   C_id: Unique user id done by auto increment.

   Name: Name of the user.

   Phone: Phone number of the user.

   Email: Email id of the user.

   Address: It stores the address of the customer

2. Category: It stores the category to which the product belong.

   Category_id: Unique category id done by auto increment.

   Category_name: It stores the category name.

3. Seller: Stores the details of the seller.

   Seller_id: Unique id for the seller.

   Seller_name: name of the seller.

   Seller_Phone: Phone number of the seller.

   Seller_Email: Email id of the seller.

   Seller_Address: It stores the address of the seller

4. Product: It stores the product details

   Pr_id : unique id that is auto incremented

   pr_name : It store the name of the produt

   pr_price: price of the products

   Quantity: quantity

   pr_status : whether the product is available or notations.

   pr_desc: description of the product.

   seller_id : foreign key referencing seller table.

   category_id : foreign key referencing category table

   pr_image: stores the image of the product

5. Cart : it stores the details of the product and quantity that is there in cart

   C_id : referencing customer id in customer table

   p_id : referencing product id in product table

   quantity : quantity of the product .

   order_date : date of the product added to the cart.

6. Orders : it stores the details of the ordered product and the customer who ordered

   C_id : referencing customer id in customer table

   p_id : referencing product id in product table

   order_no : unique id generated to every order.

Chapter 4

IMPLEMENTATION

4.1 IMPLEMENTATION TECHNOLOGY

The e-commerce website has been built using various technologies such as HTML, CSS, JavaScript, and a server-side programming language Python. These technologies can be used to create the front-end (user interface) and back-end (server-side logic) of the website. A database, such as MySQL, is also typically used to store and retrieve information about products, orders, and customers. Payment gateway integration is also an important aspect of e-commerce websites, which can be achieved using services such as PayPal or Stripe (Note:but not implemented in this project). Additionally, security measures such as SSL encryption has been implemented to protect sensitive information.

4.2 SOURCE CODE

```
class ProductView(View):

def get(self, request):

fashion = product.objects.filter(category_id='1')

electronics = product.objects.filter(category_id='2')

books = product.objects.filter(category_id='3')

grocery = product.objects.filter(category_id='4')

return render(request, 'app/home.html',{'fashion': fashion, 'electronics':electronics,'books':books,
'grocery':grocery})


class ProductDetailView(View):

def get(self, request, pk):
```

```
pro = product.objects.get(pk=pk)

return render(request, 'app/productdetail.html', {'product':pro})

def add_to_cart(request):

user=request.user

product_id = request.GET.get('prod_id')

pro = product.objects.get(pr_id=product_id)

cart(user=user, Product=pro).save()

return redirect('/cart')

def show_cart(request):

if request.user.is_authenticated:

user = request.user

Cart = cart.objects.filter(user=user)

#print(Cart)

amount = 0.0

shipping_amt = 120.0

total_amt = 0.0

cart_product = [p for p in cart.objects.all() if p.user==user]

#print(cart_product)

if cart_product:

for p in cart_product:
```

```
tempamt = (p.order_quantity * p.Product.pr_price)

amount += tempamt

total_amt = amount + shipping_amt

return render(request, 'app/addtocart.html', {'carts': Cart, 'total_amt': total_amt

, 'amount': amount})

else:

return render(request, 'app/emptycart.html')



def plus_cart(request):

if request.method == 'GET':

prod_id = request.GET['prod_id']

print(prod_id)

c = cart.objects.get(Q(Product=prod_id) & Q(user=request.user))

c.order_quantity+=1

c.save()

amount = 0.0

shipping_amt = 120.0

cart_product = [p for p in cart.objects.all() if p.user==request.user]

for p in cart_product:

tempamt = (p.order_quantity * p.Product.pr_price)
```

```
amount += tempamt

return JsonResponse(data)

def electronics(request, data=None):

if data==None:

elec =product.objects.filter(category_id='2')

elif data=='below':

elec = product.objects.filter(category_id='2').filter(pr_price__lt=10000)

elif data=='above':

elec = product.objects.filter(category_id='2').filter(pr_price__gt=10000)

return render(request, 'app/electronics.html', {'electronics':elec})

class CustomerRegistrationView(View):

def get(self, request):

form = CustomerRegistrationForm()

return render(request, 'app/customerregistration.html',

{'form':form})

def post(self, request):

form = CustomerRegistrationForm(request.POST)

if form.is_valid():

messages.success(request, 'Congratulations!! Registered Successfully')

form.save()
```

```
return render(request, 'app/customerregistration.html',{'form':form})

class ProfileView(View):

def get(self, request):

form = CustomerProfileForm()

return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary'})

def post(self, request):

form = CustomerProfileForm(request.POST)

if form.is_valid():

usr = request.user

name = form.cleaned_data['c_name']

email = form.cleaned_data['email_id']

mobile_no = form.cleaned_data['mobile_no']

customer_address = form.cleaned_data['c_add']

reg = customer(user = usr, c_name=name, email_id=email, mobile_no=mobile_no,
c_add=customer_address)

reg.save()

messages.success(request, 'Congratulations!!! Profile has been Updated Successfully')

return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary'})
```

4.3 RESULT

The resulting system is able to:

1. Authenticate user credentials during login.
2. Salted encryption for security of user passwords.
3. Register new users.
4. Allow users to view different products.
5. Allow user to see same product sold by different sellers.
6. Ability to buy products they want to.

Chapter 5

SNAPSHOTS

This chapter consists of working screenshots of the project.

5.1 LOGIN PAGE

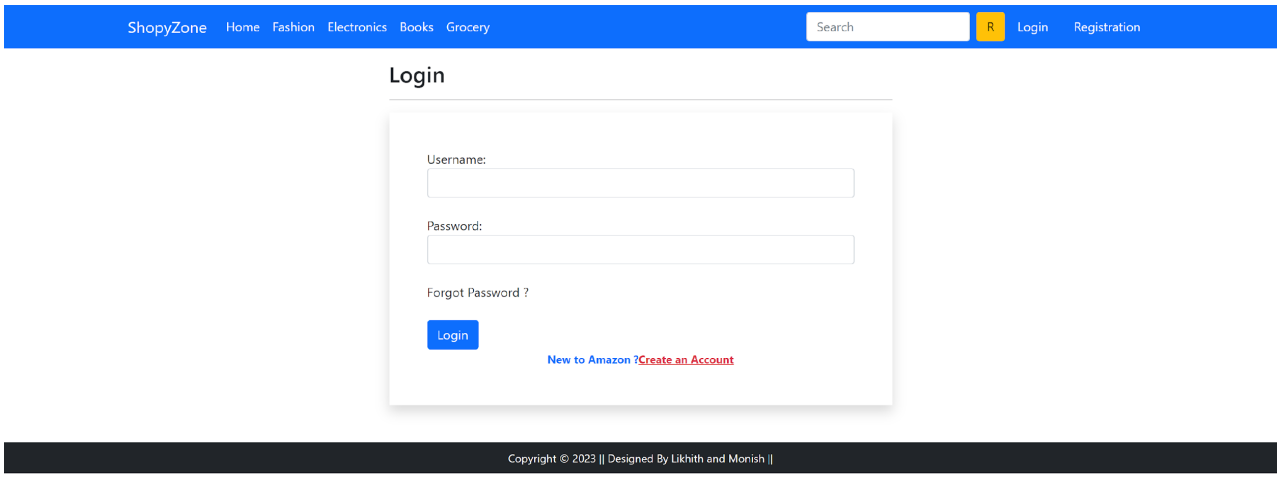

FIg . 5.1.1 Login page

This is the login page for existing users and is the first page shown to any customer.

5.2 REGISTRATION PAGE

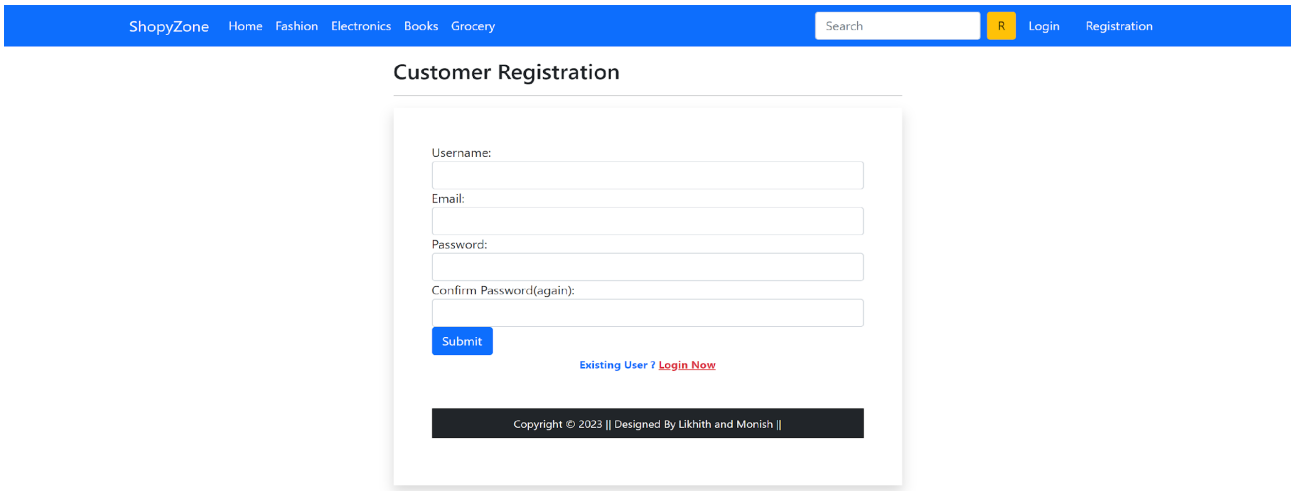

FIg . 5.1.2 Registration page

This is the registration page for any new users.

5.3 HOME PAGE

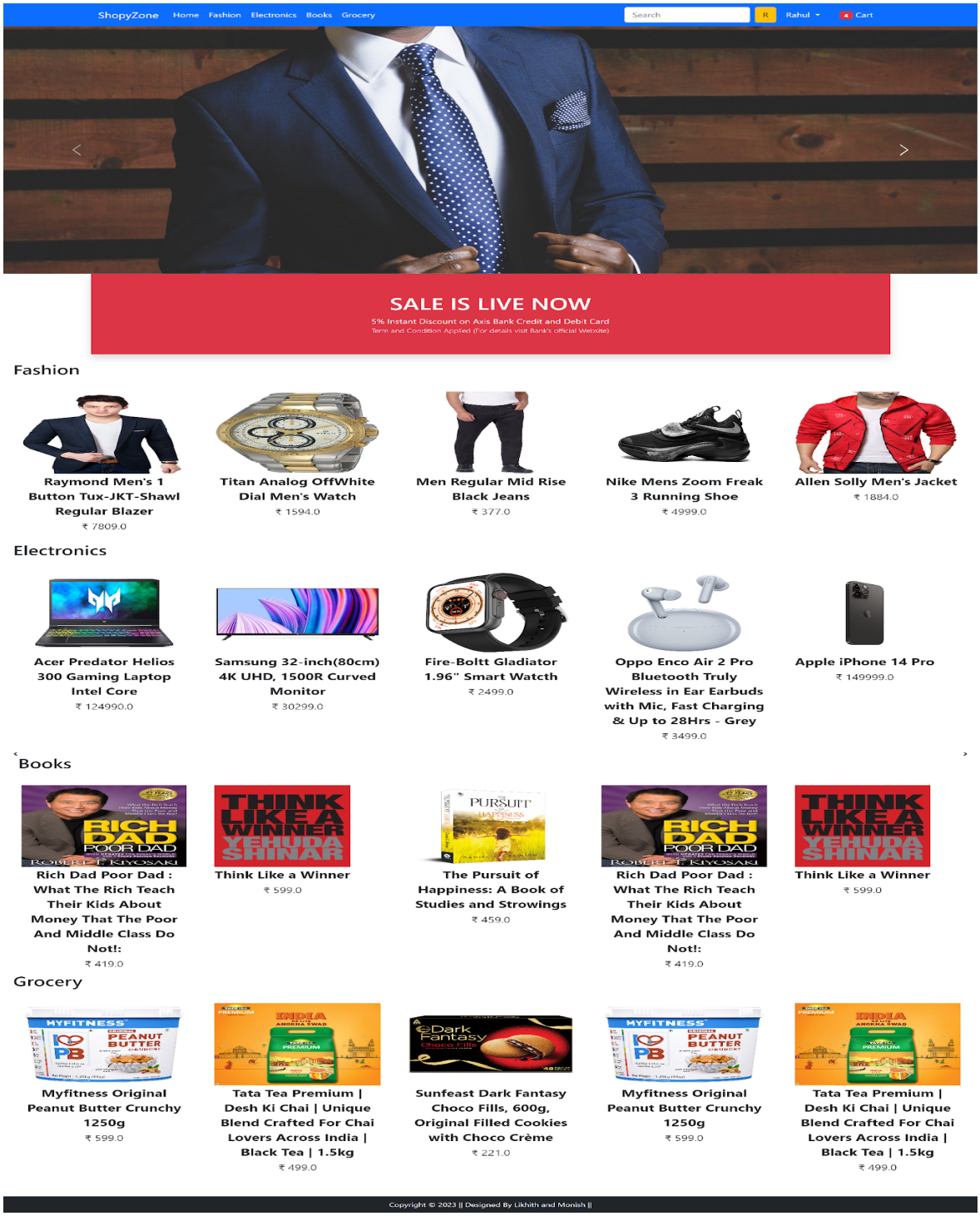

FIg . 5.1.3 Home page

First home page shown to customers after login.

5.4 LIST OF PRODUCTS IN ELECTRONICS

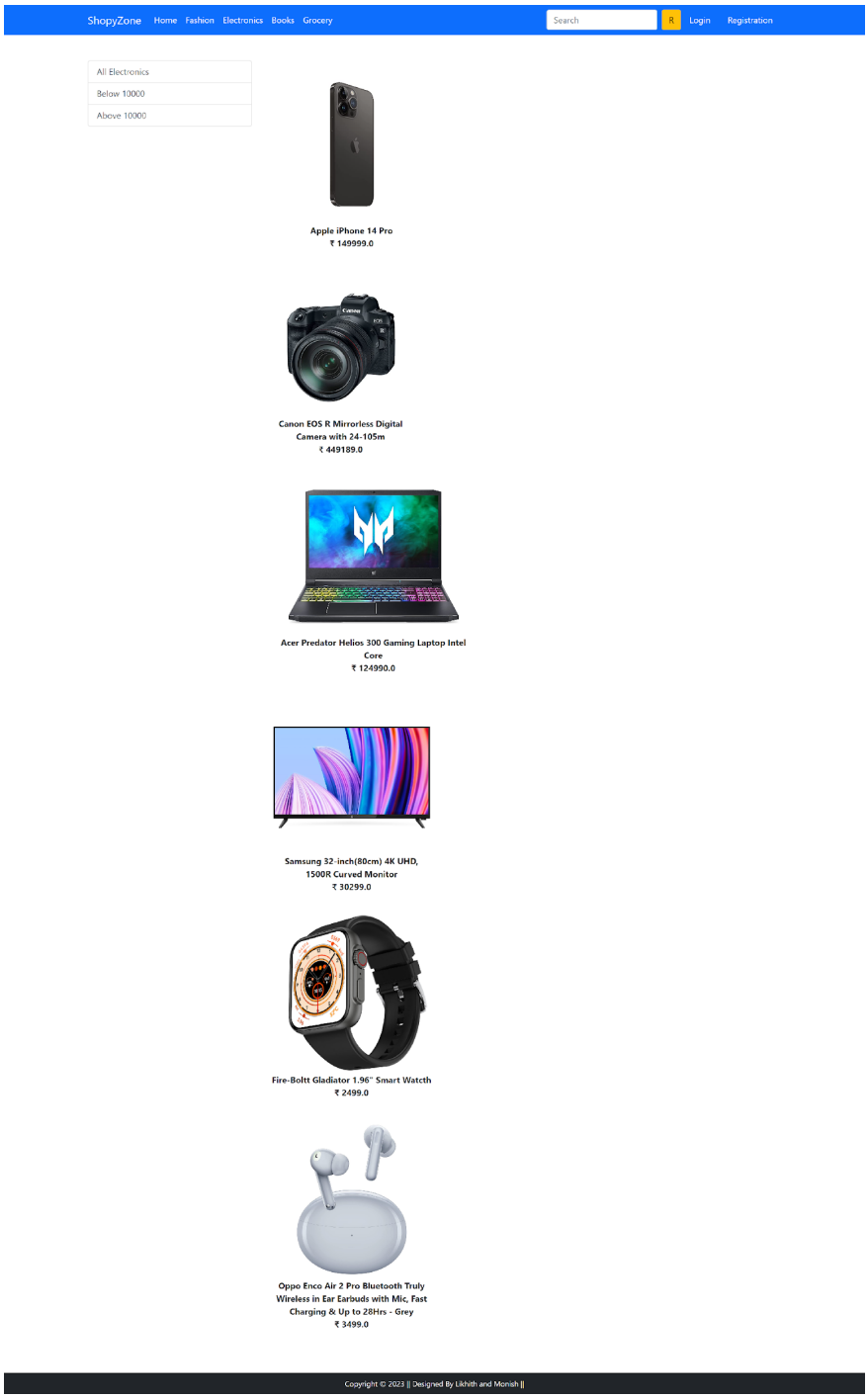

FIg . 5.1.4 Product list

List of Products in Electronics category.

5.5 PRODUCT DESCRIPTION

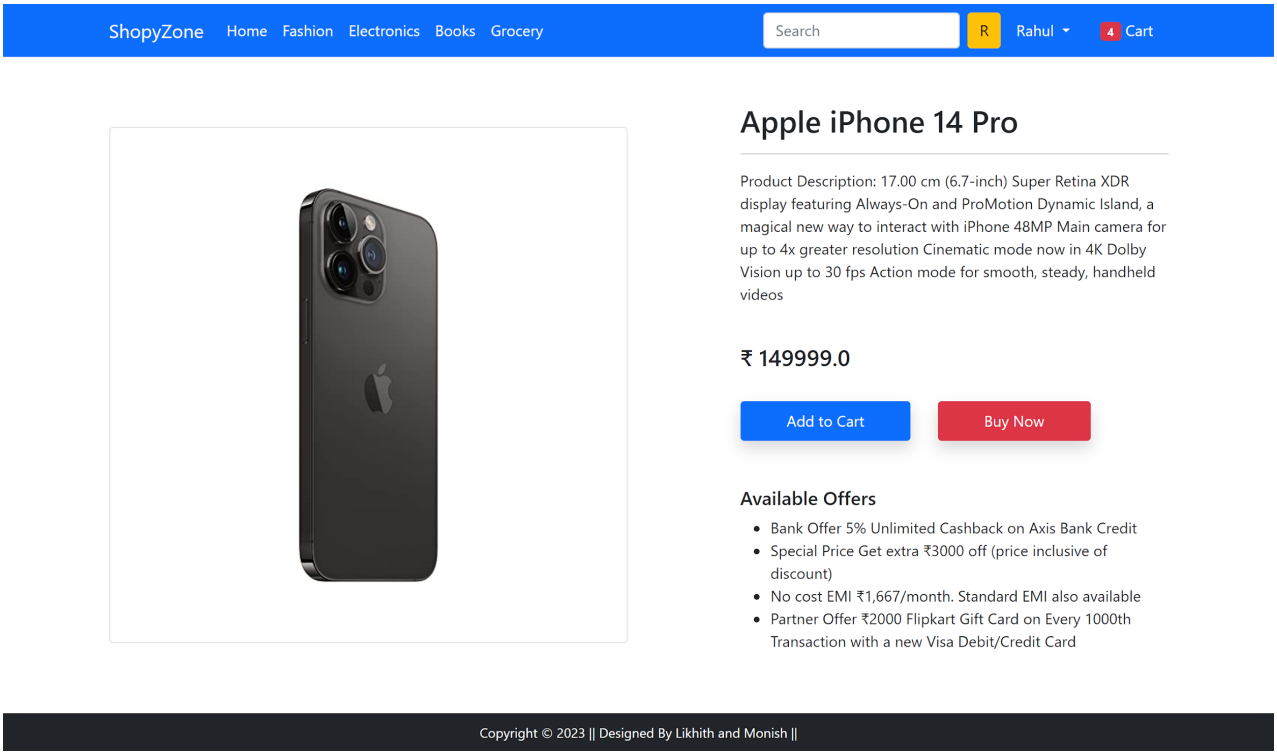

FIg . 5.1.5 Product page

This page shows the detailed description of a selected product.

5.6 CART PAGE

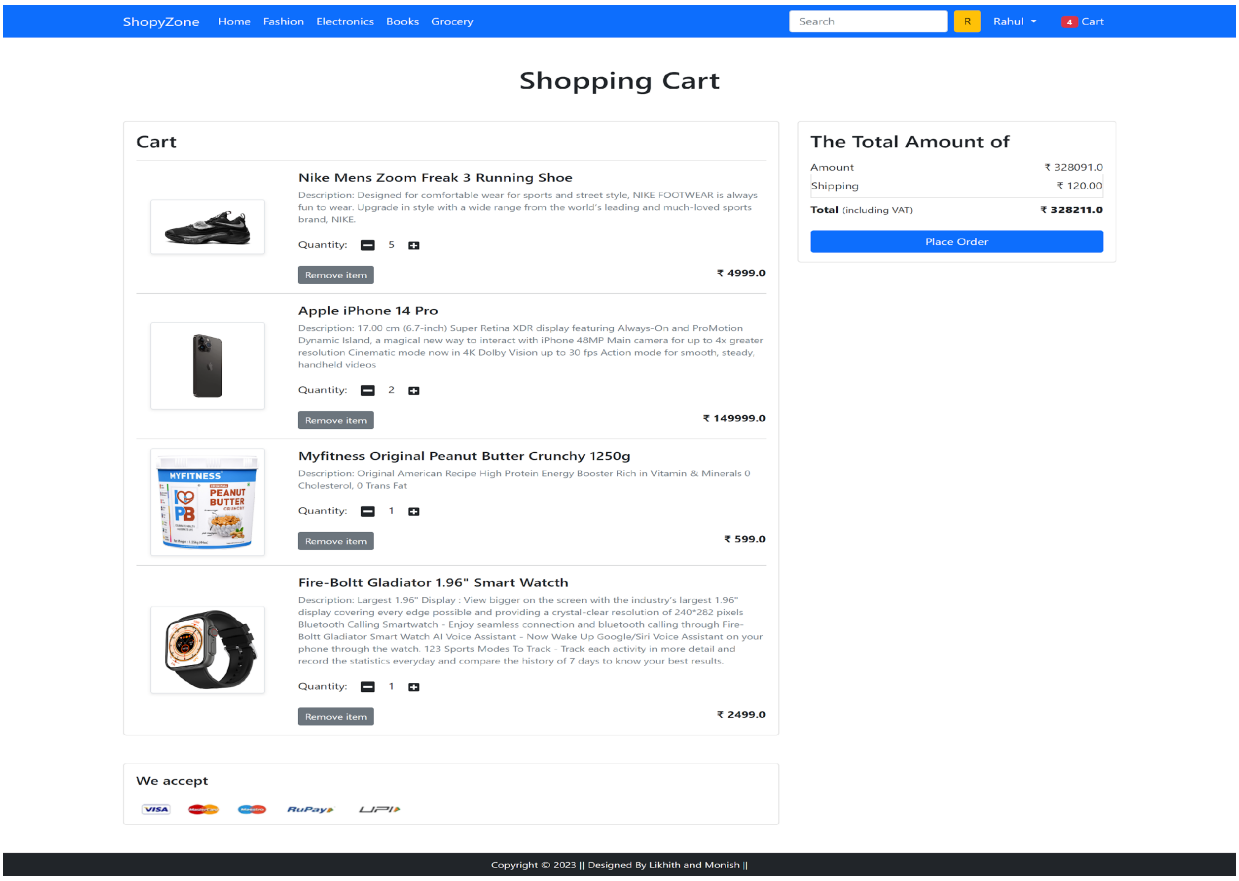

FIg . 5.1.6 Cart page

This allows users to see all the products , the total cost of all the products in his cart .

5.7 ADMIN PAGE

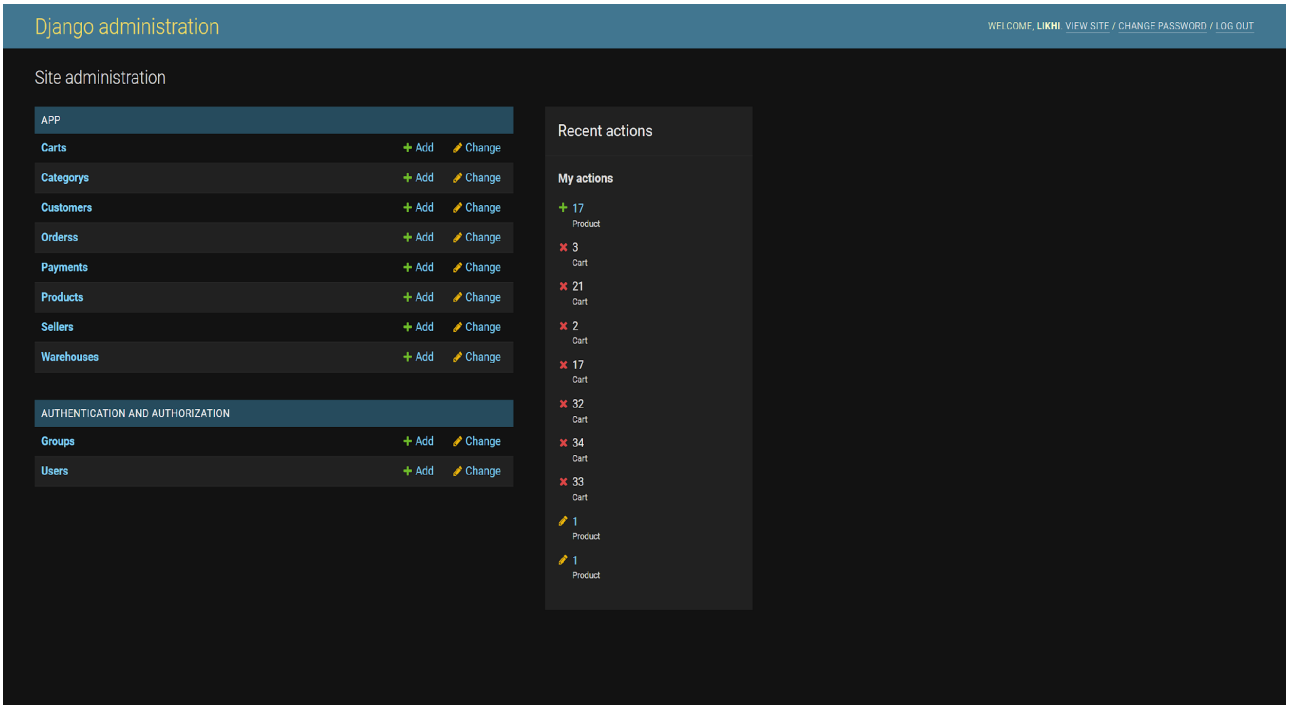

FIg . 5.1.7 Admin page

This page allows administrator to monitor and manage the system.

CONCLUSION

Utilizing a custom website exclusively for marts presents an interesting opportunity to provide the most up-to-date information about the latest products and services. By expanding the digital reach, the client can be sure to get the most out of their investment. With this increased exposure, the business stands to benefit from increased sales and better customer engagement. The website should be designed with scalability in mind, providing options for growth and future enhancements that can meet the ever-changing demands of the market.

1. Centralized database
2. Easier buying of various products.
3. User friendly environment.
4. Secure login-logout (password encryption)
5. Inexpensive
6. Fully Informative

Add On's :

The purpose of this document is to outline our plans for the future development of a custom website specifically for marts. We plan to optimize the user experience by making use of the latest technology, such as chatbots and AI-powered personalization. Moreover, we will be ensuring that the website is secure and has all the necessary features to make it easy to use and navigate. We also plan to introduce new features and functions, such as loyalty programs, to further enhance the user experience.

FUTURE ENHANCEMENTS

Future upgrades to this project will implement:

Better interfaces for the ability to view of various products including better description, more data across various products.

1. AI-powered personalization etc.
2. ChatBot , Product search through a photo click etc.
3. Better banking implementations between the customer and his bank.
4. Product rating and customer reviews system.
5. More payment methods

REFERENCES

1. Ramakrishnan, R., & Gehrke, J. (2011). Database management systems. Boston: McGraw-Hill.

2. Monson-Haefel, R. (2007). J2EE Web services. Boston, Mass: Addison-Wesley. Silberschatz A., Korth H. F., & Sudarshan S. (2011).

3. Database systems concepts. Estados Unidos: McGraw-Hill Companies, Inc.

4. Hanna P. (2002): JSP 2.0 The Complete Reference, Second Edition McGraw Hill Education.

5. https://www.w3schools.com

6. https://getbootstrap.com/