RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

ONLINE SHOPPING CART

A MINI PROJECT REPORT

Submitted by

ARUL JOTHI P 230701034

DEEPA S 230701065

In partial fulfillment for the award of the degree of

BACHELOR OF

ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM

CHENNAI-602105
2024 - 2025

BONAFIDE CERTIFICATE

Certified that this project report “**ONLINE SHOPPING CART**” is the Bonafide work of **“ARUL JOTHI P (230701034), DEEPA S (230701065)”** who carried out the project work under my supervision.

Submitted for the Practical Examination held on

SIGNATURE

Mrs.Deepa B,
Assistant Professor (SS)
CSE,
Rajalakshmi Engineering College,(Autonomous),
Thandalam, Chennai - 602 105

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The **Online Shopping Cart** is a web-based application designed to provide a seamless e-commerce experience for customers and administrators. Built using Java, JDBC, Servlets, and MySQL, this system facilitates online shopping by allowing customers to browse products, add items to their cart, and complete purchases, while also providing administrators with tools to manage inventory, pricing, and orders.

The customer portal allows users to register accounts, view available products, and manage their shopping cart before proceeding to checkout. Secure payment processing ensures that transactions are completed safely. The admin portal, on the other hand, allows administrators to add, update, and delete product listings, adjust prices, and monitor stock levels, ensuring that inventory is efficiently managed.

The back-end of the system leverages JDBC to interact with the MySQL database, handling customer orders, product details, and transaction histories. The front-end is developed using HTML, CSS, JavaScript, and Bootstrap, providing a responsive and intuitive user interface that works across devices.

This online shopping cart application demonstrates the integration of Java web technologies to build a fully functional e-commerce solution, incorporating key features such as user authentication, secure order processing, and inventory management.

TABLE OF CONTENTS

I. INTRODUCTION

1.1 INTRODUCTION

In today's digital era, online shopping has become a fundamental aspect of the retail experience, offering unparalleled convenience for consumers. The Online Shopping Cart project is a web-based solution designed to deliver a seamless shopping experience for customers, while also providing effective inventory and order management tools for administrators. Built using core Java web technologies such as JDBC and Servlets, the system ensures a secure, scalable, and user-friendly platform for both buyers and sellers.

The application is divided into two primary interfaces: the customer portal and the admin panel. The customer portal allows users to browse products, add items to their cart, and securely complete their purchases, offering an intuitive shopping experience. Meanwhile, the admin panel provides tools for administrators to manage products, update pricing, monitor stock levels, and track customer orders, ensuring smooth business operations.

Using modern front-end technologies like HTML5, CSS3, JavaScript, and Bootstrap, the system guarantees a responsive design that adapts seamlessly across devices, from desktop computers to mobile phones. With JDBC for database connectivity and Java Servlets for handling backend logic, the application integrates database management with essential e-commerce features such as user authentication, cart management, and secure payment processing.

This project showcases the practical application of web development and database management principles, providing a complete e-commerce solution that meets the needs of both customers and administrators in a dynamic online shopping environment.

1.2 OBJECTIVES

1. Primary Objectives

- **User-friendly platform** : Create an intuitive online shopping cart for easy book browsing and purchasing.
- **Secure inventory management** : Develop a system to track inventory, prevent overselling, and protect user data.
- **Efficient authentication** : Implement secure user login and role-based access control for customers and admins.
- **Automated transactions** : Streamline payment processing and generate digital receipts automatically.

2. Business Objectives

- **Streamline operations** : Automate inventory and sales processes to save time and reduce manual effort.
- **Reduce manual inventory management** : Use real-time updates to ensure accurate stock levels and prevent errors.
- **Increase accessibility** : Make the platform accessible across devices, providing a convenient shopping experience.
- **Maintain accurate records** : Automatically track and store sales data for reporting and financial analysis.

1.3 MODULES

1. Admin Module

- Login/Authentication: Product Management
- Inventory Control

2. User Module

- Registration/Login
- Product Browsing
- Shopping Cart
- Purchase Processing

3. Database Module

- User Data Management
- Product Inventory Records
- Transaction History
- Order Processing

4. Security Module

- User Authentication
- Session Management
- Secure Data Trans

II. SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION

ECLIPSE

Eclipse is written mostly in Java and its primary use is for developing Java applications,Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second most-popular IDE for Java development, and, until 2016, was the most popular. Eclipse is written mostly in Java and its primary use is for developing Java applications.

Eclipse Enterprise Edition (EE) is a package for developers who work with Java and web applications. It includes tools for:

- Java
- JavaScript
- TypeScript
- JavaServer Pages and Faces
- Web Services
- Maven and Gradle
- Git

Eclipse EE is a version of Eclipse that comes with tools to make it easier to write server code. For example, you can compile and run a server by pressing the play button.

2.2 LANGUAGES

2.2.1 MySQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

2.2.2 JAVA

Java is a set of computer software and specifications that provides a software platform for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. Java applets, which are less common than standalone Java applications, were commonly run in secure, sandboxed environments to provide many features of native applications through being embedded in HTML pages.

2.2.3 HTML

HTML, or Hypertext Markup Language, is the standard language used to create web pages. It defines the structure and content of web documents using tags and attributes to format text, embed images, create links, and build interactive elements. HTML facilitates communication between web browsers and servers, making it a crucial skill for web developers.

HTML was invented by Tim Berners-Lee, a physicist at CERN, in 1990. His goal was to create a simple way to share and access documents over the Internet. Since its inception, HTML has evolved significantly, becoming the foundation of web development. When working with HTML, you use a simple code structure that includes tags and attributes to build the layout of a webpage.

2.2.4 CSS

CSS, which stands for Cascading Style Sheets, is a language in web development that enhances the presentation of HTML elements. By applying styles like color, layout, and spacing, CSS makes web pages visually appealing and responsive to various screen sizes. CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility, since the content can be written without concern for

its presentation; 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, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

2.2.5 JAVASCRIPT

JavaScript, often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

III. REQUIREMENTS AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION

User Requirements

The system requirement in the online bookstore management focuses on the ability to search for books by title, author, or genre by the customer. It also includes user account management, viewing book details, and making purchases.

System Requirements

There should be a database backup of the online bookstore system. The operating system should be Windows 10 or a higher version of Windows.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

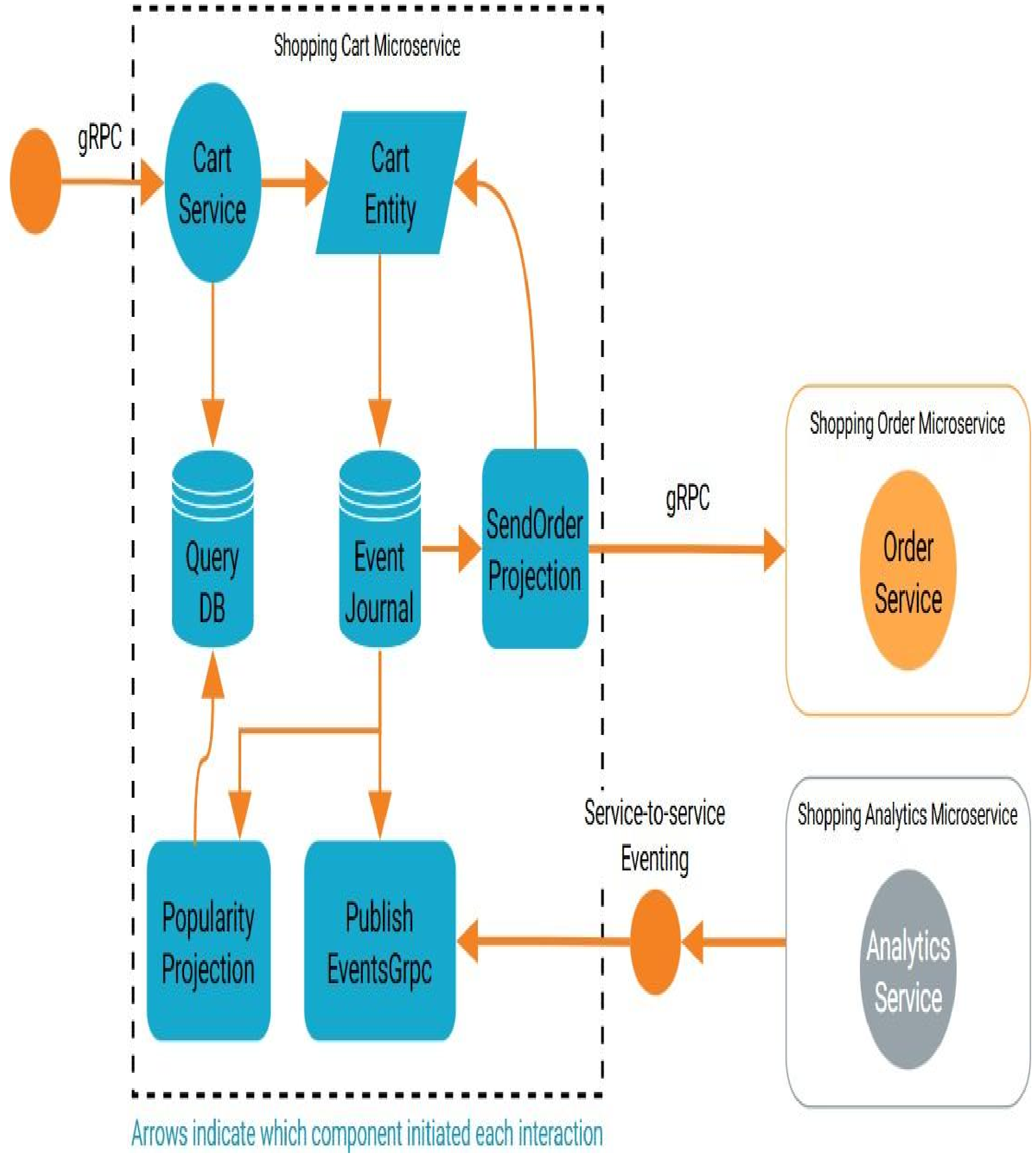
Software Requirements

- Operating System: Windows 10
- Front End: HTML, CSS, JavaScript
- Back End: Java, MySQL

Hardware Requirements

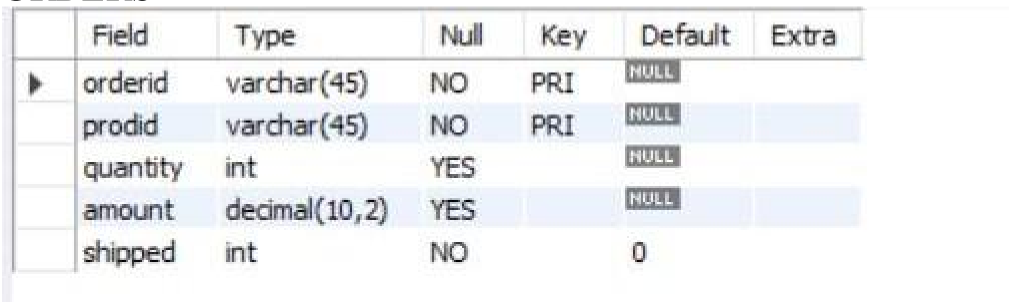
- Desktop PC or Laptop
- Printer (optional)
- Operating System: Windows 10
- Intel® Core™ i3-6006U CPU @ 2.00GHz or higher
- 4.00 GB RAM or higher
- 64-bit operating system, x64 based processor
- Monitor Resolution: 1024 x 768 or higher
- Keyboard and Mouse

ONLINE SHOPPING CART ARCHITECTURE


Shopping Cart Microservice
gRPC
Cart Service
Cart Entity
Query DB
Event Journal
SendOrder Projection
gRPC
Shopping Order Microservice
Order Service
Popularity Projection
Publish EventsGrpc
Service-to-service Eventing
Shopping Analytics Microservice
Analytics Service
Arrows indicate which component initiated each interaction

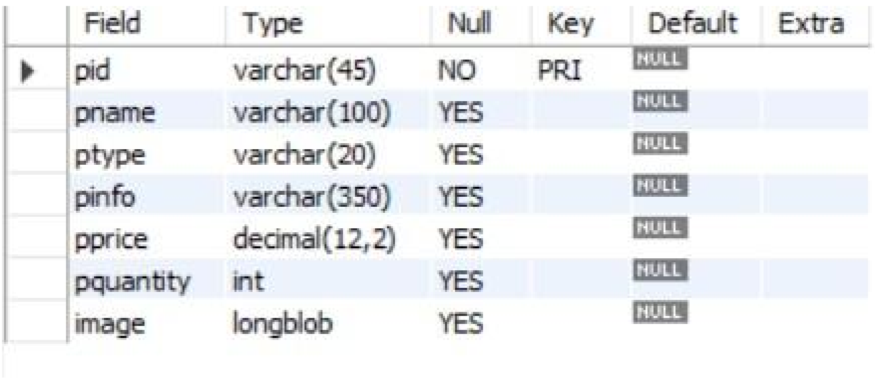
3.4 DATA DICTIONARY

ORDERS



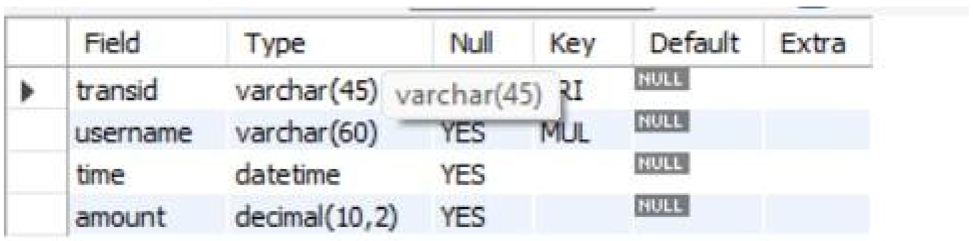
Field	Type	Null	Key	Default	Extra
orderid	varchar(45)	NO	PRI	NULL	
prodid	varchar(45)	NO	PRI	NULL	
quantity	int	YES		NULL	
amount	decimal(10,2)	YES		NULL	
shipped	int	NO		0	

PRODUCT



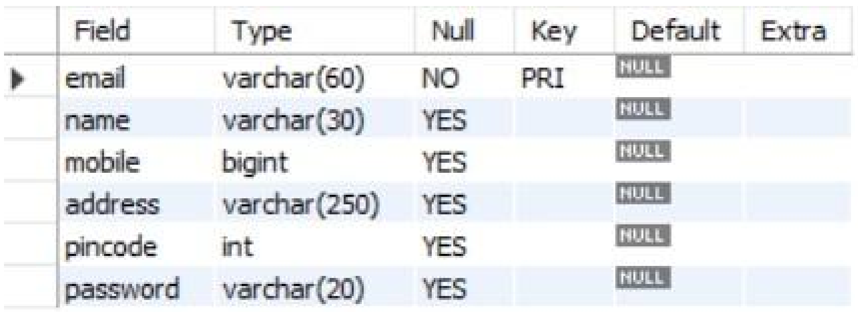
Field	Type	Null	Key	Default	Extra
pid	varchar(45)	NO	PRI	NULL	
pname	varchar(100)	YES		NULL	
ptype	varchar(20)	YES		NULL	
pinfo	varchar(350)	YES		NULL	
pprice	decimal(12,2)	YES		NULL	
pquantity	int	YES		NULL	
image	longblob	YES		NULL	

TRANSACTIONS



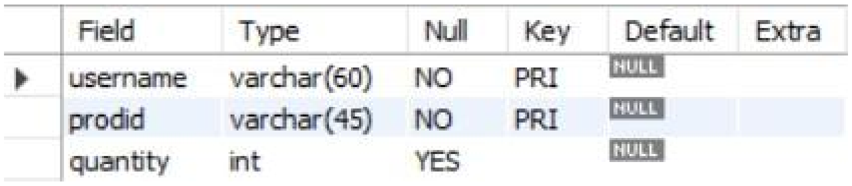
Field	Type	Null	Key	Default	Extra
transid	varchar(45)	varchar(45)	RI	NULL	
username	varchar(60)	YES	MUL	NULL	
time	datetime	YES		NULL	
amount	decimal(10,2)	YES		NULL	

USER



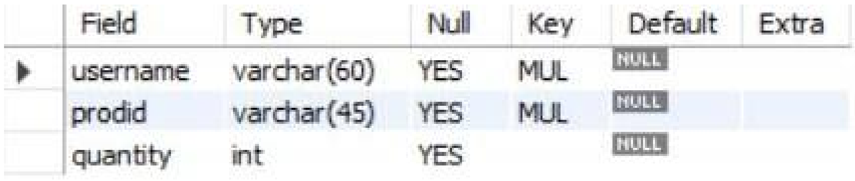
Field	Type	Null	Key	Default	Extra
email	varchar(60)	NO	PRI	NULL	
name	varchar(30)	YES		NULL	
mobile	bigint	YES		NULL	
address	varchar(250)	YES		NULL	
pincode	int	YES		NULL	
password	varchar(20)	YES		NULL	

USER_DEMAND



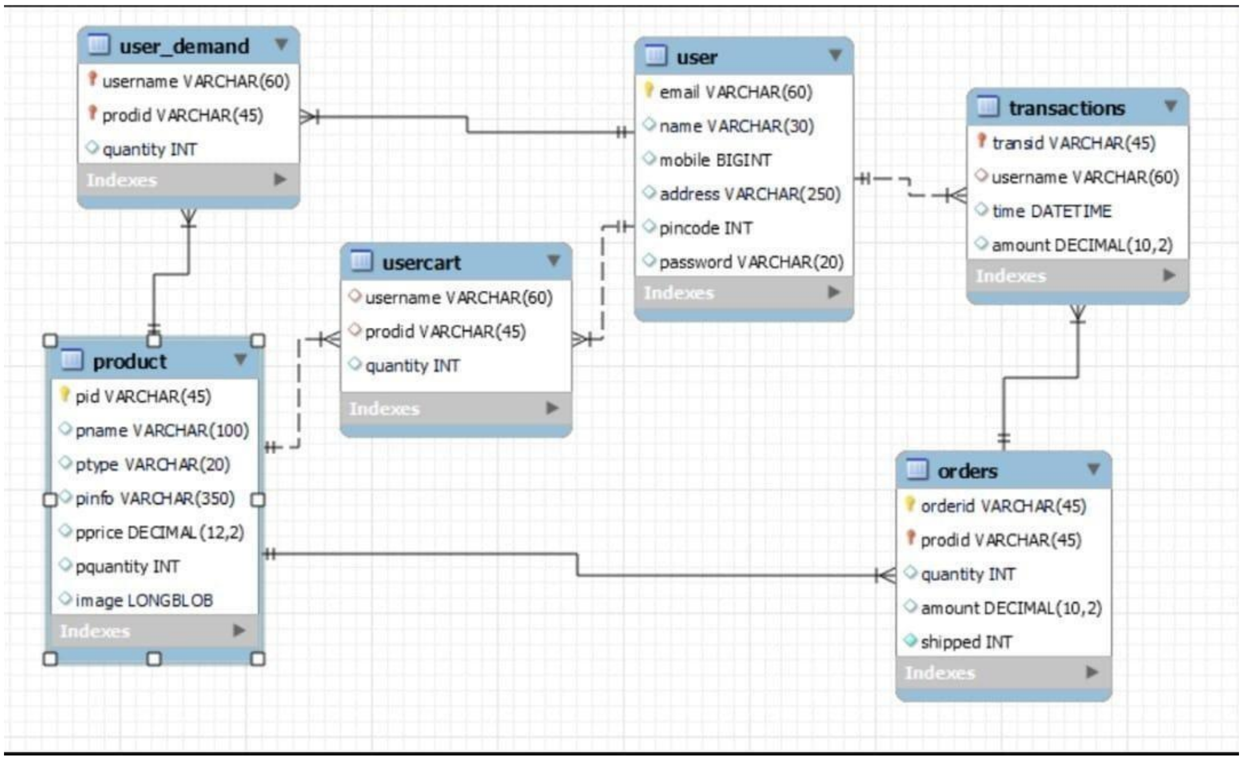
Field	Type	Null	Key	Default	Extra
username	varchar(60)	NO	PRI	NULL	
prodid	varchar(45)	NO	PRI	NULL	
quantity	int	YES		NULL	

USERCART



Field	Type	Null	Key	Default	Extra
username	varchar(60)	YES	MUL	NULL	
prodid	varchar(45)	YES	MUL	NULL	
quantity	int	YES		NULL	

3.5 ER DIAGRAM



IV. PROGRAM CODE

DATABASE

- CREATE TABLE IF NOT EXISTS `shopping-cart`.`product` (

```
  `pid` VARCHAR(45) NOT NULL,

  `pname` VARCHAR(100) NULL DEFAULT
NULL,

  `ptype` VARCHAR(20) NULL DEFAULT
NULL,

  `pinfo` VARCHAR(350) NULL DEFAULT
NULL,

  `pprice` DECIMAL(12,2) NULL DEFAULT
NULL,

  `pquantity` INT NULL DEFAULT NULL,

  `image` LONGBLOB NULL DEFAULT
NULL,

  PRIMARY KEY (`pid`))
```

APPLICATION PROPERTIES

```
db.driverName = com.mysql.cj.jdbc.Driver

db.connectionString = jdbc:mysql://localhost:3306/shopping-cart

db.username = root

db.password = root

mailer.email=your_email

mailer.password=your_app_password_generated_from_email
```

CART SERVICE

```
import java.util.List;

import com.shashi.beans.CartBean;

public interface CartService {

          public String addProductToCart(String userId, String prodId, int prodQty);

           public String updateProductToCart(String userId, String prodId, int prodQty);

           public List<CartBean> getAllCartItems(String userId);

           public int getCartCount(String userId);

           public int getCartItemCount(String userId, String itemId);

           public String removeProductFromCart(String userId, String prodId);

           public boolean removeAProduct(String userId, String prodId);

}
```

DEMAND SERVICE

```
  package com.shashi.service;

   import java.util.List;

   import com.shashi.beans.DemandBean;

    public interface DemandService {
```

```
    public boolean addProduct(String userId,String prodId,int demandQty);
    public boolean addProduct(DemandBean userDemandBean);
    public boolean removeProduct(String userId,String prodId);
    public List<DemandBean> haveDemanded(String prodId);
}
```

ADD TO CART

```
package com.shashi.srv;


import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.shashi.beans.DemandBean;
import com.shashi.beans.ProductBean;
import com.shashi.service.impl.CartServiceImpl;
import com.shashi.service.impl.DemandServiceImpl;
import com.shashi.service.impl.ProductServiceImpl;


/**
 * Servlet implementation class AddtoCart
 */
@WebServlet("/AddtoCart")
public class AddtoCart extends HttpServlet
   { private static final long
   serialVersionUID=1L;

   public AddtoCart()
      {super();
   }

   protected void doGet(HttpServletRequest request,HttpServletResponse response)
         throws ServletException,IOException {

      HttpSession session=request.getSession();
      String userName=(String)session.getAttribute("username");
      String password=(String)session.getAttribute("password");
      String usertype=(String)session.getAttribute("usertype");

if(userName==null||password==null||usertype==null||!usertype.equalsIgnoreCase("customer
")) {
         response.sendRedirect("login.jsp?message=Session Expired,Login Again to
Continue!");
         return;
      }

      String userId=userName;
      String prodId=request.getParameter("pid");
```

```
int pQty=Integer.parseInt(request.getParameter("pqty"));
```

```
        CartServiceImpl cart=new CartServiceImpl();

        ProductServiceImpl productDao=new ProductServiceImpl();

        ProductBean product=productDao.getProductDetails(prodId);

        int availableQty=product.getProdQuantity();

        int cartQty=cart.getProductCount(userId,prodId);

        pQty+=cartQty;


        PrintWriter pw=response.getWriter();

        response.setContentType("text/html");

        if(pQty==cartQty) {

            String status=cart.removeProductFromCart(userId,prodId);

            RequestDispatcher rd=request.getRequestDispatcher("userHome.jsp");

            rd.include(request,response);

pw.println("<script>document.getElementById('message').innerHTML='"+status+"'</script
>");

        } else if(availableQty<pQty)

            {String status=null;

            if(availableQty==0) {

                status="Product is Out of Stock!";

            } else {

                cart.updateProductToCart(userId,prodId,availableQty);

                status="Only "+availableQty+" no of "+product.getProdName()+" are available in
the shop! So we are adding only "+availableQty+" products into Your Cart";

            }

            DemandBean demandBean=new DemandBean(userName,product.getProdId(),pQty-
availableQty);
```

```
            DemandServiceImpl demand=new DemandServiceImpl();

            boolean flag=demand.addProduct(demandBean);

            if(flag)

               status+="<br/>Later, We Will Mail You when "+product.getProdName()+" will
be available into the Store!";

            RequestDispatcher rd=request.getRequestDispatcher("cartDetails.jsp");

            rd.include(request,response);


pw.println("<script>document.getElementById('message').innerHTML='"+status+"'</script
>");

        } else {

            String status=cart.updateProductToCart(userId,prodId,pQty);

            RequestDispatcher rd=request.getRequestDispatcher("userHome.jsp");

            rd.include(request,response);


pw.println("<script>document.getElementById('message').innerHTML='"+status+"'</script
>");

        }

    }


    protected void doPost(HttpServletRequest request,HttpServletResponse response)

            throws ServletException,IOException {

        doGet(request,response);

    }

}
```

LOGIN

```
package com.shashi.srv;


import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import com.shashi.beans.UserBean;

import com.shashi.service.impl.UserServiceImpl;


/**

 * Servlet implementation class LoginSrv

 */

@WebServlet("/LoginSrv")

public class LoginSrv extends HttpServlet

   { private static final long

   serialVersionUID=1L;


   public LoginSrv()

      {super();

   }
```

```
protected void doGet(HttpServletRequest request,HttpServletResponse response)
```

```
    throws ServletException,IOException {

  String userName=request.getParameter("username");
  String password=request.getParameter("password");
  String userType=request.getParameter("usertype");
  response.setContentType("text/html");

  String status="Login Denied! Invalid Username or password.";

  if(userType.equals("admin")) { // Login as Admin
    if(password.equals("admin")&&userName.equals("admin@gmail.com")) {
      // valid
      RequestDispatcher rd=request.getRequestDispatcher("adminViewProduct.jsp");
      HttpSession session=request.getSession();
      session.setAttribute("username",userName);
      session.setAttribute("password",password);
      session.setAttribute("usertype",userType);
      rd.forward(request,response);
    } else {
      // Invalid
      RequestDispatcher
rd=request.getRequestDispatcher("login.jsp?message="+status);
      rd.include(request,response);
    }
  } else { // Login as customer
    UserServiceImpl udao=new UserServiceImpl();
```

```
            status=udao.isValidCredential(userName,password);
            if(status.equalsIgnoreCase("valid")) {
                // valid user
                UserBean user=udao.getUserDetails(userName,password);
                HttpSession session=request.getSession();
                session.setAttribute("userdata",user);
                session.setAttribute("username",userName);
                session.setAttribute("password",password);
                session.setAttribute("usertype",userType);
                RequestDispatcher rd=request.getRequestDispatcher("userHome.jsp");
                rd.forward(request,response);
            } else {
                // invalid user
                RequestDispatcher
rd=request.getRequestDispatcher("login.jsp?message="+status);
                rd.forward(request,response);
            }
        }
    }

    protected void doPost(HttpServletRequest request,HttpServletResponse response)
            throws ServletException,IOException {
        doGet(request,response);
    }
}
```

LOGOUT

```
package com.shashi.srv;

import java.io.IOException;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


/**

 * Servlet implementation class LogoutSrv

 */

@WebServlet("/LogoutSrv")

public class LogoutSrv extends HttpServlet

   { private static final long

   serialVersionUID=1L;


   public LogoutSrv()

      {super();

   }


   protected void doGet(HttpServletRequest request,HttpServletResponse response)

         throws ServletException,IOException {
```

```
response.setContentType("text/html");
```

```
        HttpSession session=request.getSession();


        session.setAttribute("username",null);
        session.setAttribute("password",null);
        session.setAttribute("usertype",null);
        session.setAttribute("userdata",null);


        RequestDispatcher rd=request.getRequestDispatcher("login.jsp?message=Successfully
Logged Out!");


        rd.forward(request,response);
    }


    protected void doPost(HttpServletRequest request,HttpServletResponse response)
            throws ServletException,IOException {


        doGet(request,response);
    }
}
```

UPDATE PRODUCT

```
package com.shashi.srv;


import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.shashi.beans.ProductBean;
import com.shashi.service.impl.ProductServiceImpl;

@WebServlet("/UpdateProductSrv")
public class UpdateProductSrv extends HttpServlet
  {private static final long serialVersionUID=1L;

  public UpdateProductSrv()
    {super();
  }

  protected void doGet(HttpServletRequest request,HttpServletResponse response)
        throws ServletException,IOException {

    HttpSession session=request.getSession();
    String userType=(String)session.getAttribute("usertype");
    String userName=(String)session.getAttribute("username");
    String password=(String)session.getAttribute("password");

    if(userType==null||!userType.equals("admin")) {
```

```
    response.sendRedirect("login.jsp?message=Access Denied, Login As Admin!!");
    return;

} else if(userName==null||password==null) {

    response.sendRedirect("login.jsp?message=Session Expired, Login Again!!");
    return;
}

// Login success

String prodId=request.getParameter("pid");
String prodName=request.getParameter("name");
String prodType=request.getParameter("type");
String prodInfo=request.getParameter("info");
Double prodPrice=Double.parseDouble(request.getParameter("price"));
Integer prodQuantity=Integer.parseInt(request.getParameter("quantity"));

ProductBean product=new ProductBean();
product.setProdId(prodId);
product.setProdName(prodName);
product.setProdInfo(prodInfo);
product.setProdPrice(prodPrice);
product.setProdQuantity(prodQuantity);
product.setProdType(prodType);
```

```
        ProductServiceImpl dao=new ProductServiceImpl();


        String status=dao.updateProductWithoutImage(prodId,product);


        RequestDispatcher
rd=request.getRequestDispatcher("updateProduct.jsp?prodid="+prodId+"&message="+statu
s);

        rd.forward(request,response);


    }


    protected void doPost(HttpServletRequest request,HttpServletResponse response)

            throws ServletException,IOException {


        doGet(request,response);

    }


}
```

IDUtil

```
package com.shashi.utility;


import java.text.SimpleDateFormat;

import java.util.Date;
```

```
public class IDUtil {


    public static String generateId()
        {String pId = null;


        SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMddhhmmss");
        pId = sdf.format(new Date());
        pId = "P" + pId;


        return pId;
    }


    public static String generateTransId()
        {String tId = null;


        SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMddhhmmss");
        tId = sdf.format(new Date());
        tId = "T" + tId;


        return tId;
    }
}
```

JAVA MAIL

```
package com.shashi.utility;
```

```
import java.util.Properties;
import java.util.ResourceBundle;
import java.util.logging.Level;
import java.util.logging.Logger;

import jakarta.mail.Authenticator;
import jakarta.mail.Message;
import jakarta.mail.MessagingException;
import jakarta.mail.PasswordAuthentication;
import jakarta.mail.Session;
import jakarta.mail.Transport;
import jakarta.mail.internet.InternetAddress;
import jakarta.mail.internet.MimeMessage;

public class JavaMailUtil {
	public static void sendMail(String recipientMailId) throws MessagingException {

		System.out.println("Preparing to send Mail");
		Properties properties = new Properties();
		String host = "smtp.gmail.com";
		properties.put("mail.smtp.host", host);
		properties.put("mail.transport.protocol", "smtp");
		properties.put("mail.smtp.auth", "true");
		properties.put("mail.smtp.starttls.enable", "true");
		properties.put("mail.smtp.port", "587");
```

```
        ResourceBundle rb = ResourceBundle.getBundle("application");

        String emailId = rb.getString("mailer.email");
        String passWord = rb.getString("mailer.password");

        properties.put("mail.user", emailId);
        properties.put("mail.password", passWord);

        Session session = Session.getInstance(properties, new Authenticator() {

            @Override
            protected PasswordAuthentication getPasswordAuthentication()
                {return new PasswordAuthentication(emailId, passWord);
            }

        });

        Message message = prepareMessage(session, emailId, recipientMailId);

        Transport.send(message);

        System.out.println("Message Sent Successfully!");

    }
```

```
        private static Message prepareMessage(Session session, String myAccountEmail,
String recipientEmail) {


                try {


                        Message message = new MimeMessage(session);


                        message.setFrom(new InternetAddress(myAccountEmail));

                        message.setRecipient(Message.RecipientType.TO, new
InternetAddress(recipientEmail));

                        message.setSubject("Welcome to Ellison Electronics");

                        message.setText("Hey! " + recipientEmail + ", Thanks  for Signing Up
with us!");

                        return message;


                } catch (Exception exception)

                        { Logger.getLogger(JavaMailUtil.class.getName()).log(Level.SEVERE,
null, exception);

                }

                return null;


        }


        protected static void sendMail(String recipient, String subject, String
htmlTextMessage) throws MessagingException {


                System.out.println("Preparing to send Mail");

                Properties properties = new Properties();
        String host = "smtp.gmail.com";

        properties.put("mail.smtp.host", host);

        properties.put("mail.transport.protocol", "smtp");

        properties.put("mail.smtp.auth", "true");

        properties.put("mail.smtp.starttls.enable", "true");

        properties.put("mail.smtp.port", "587");


        ResourceBundle rb = ResourceBundle.getBundle("application");


        String emailId = rb.getString("mailer.email");

        String passWord = rb.getString("mailer.password");


        properties.put("mail.user", emailId);

        properties.put("mail.password", passWord);


        Session session = Session.getInstance(properties, new Authenticator() {


            @Override

            protected PasswordAuthentication getPasswordAuthentication()

                {return new PasswordAuthentication(emailId, passWord);

            }


        });


        Message message = prepareMessage(session, emailId, recipient, subject,
htmlTextMessage);
```

```
        Transport.send(message);

        System.out.println("Message Sent Successfully!");

    }

    private static Message prepareMessage(Session session, String myAccountEmail,
String recipientEmail, String subject,
            String htmlTextMessage) {

        try {

            Message message = new MimeMessage(session);

            message.setFrom(new InternetAddress(myAccountEmail));
            message.setRecipient(Message.RecipientType.TO, new
InternetAddress(recipientEmail));
            message.setSubject(subject);
            message.setContent(htmlTextMessage, "text/html");
            return message;

        } catch (Exception exception)
            { Logger.getLogger(JavaMailUtil.class.getName()).log(Level.SEVERE,
null, exception);
        }
        return null;
```

```
		}

}
```

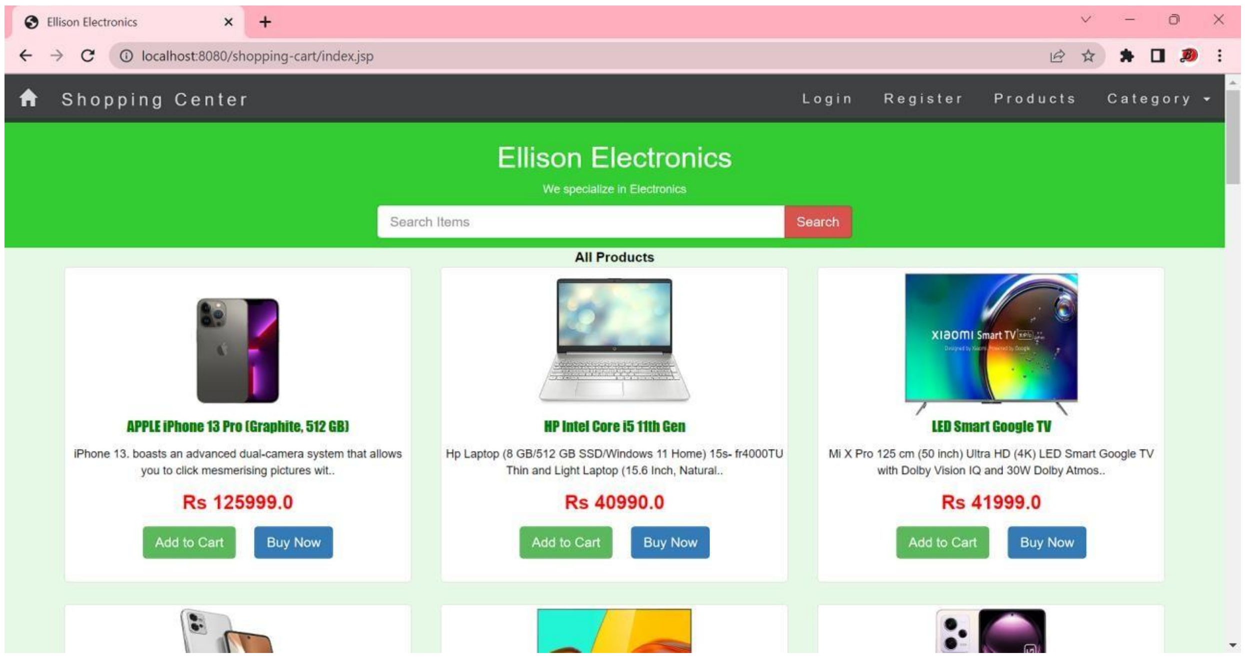
TEST MAIL

```
package com.shashi.utility;

import jakarta.mail.MessagingException;

public class TestMail {

	public static void main(String[] args)

		{try {

			String recipient = "ellison.alumni@gmail.com";

			String subject = "Mail Configuration Successfull";

			String htmlTextMessage = "" + "<html>"

					+ "<head><title>Java Mail Configuration
Test</title><style>.greenText{color:green;} p{font-size:14;}</style></head><body>"

					+ "<h2 style='color:red;'>Welcome to Ellison
Electronics</h2>" + "<p>Hey,<br>"

					+ "Thanks for singing up with Ellison Electronics.<br>"

					+ "We are glad that you choose <bold>us. We invite
you to check out our latest collection of new electonics appliances."

					+ "<br>We are providing upto 60% OFF on most of the
electronic gadgets. So please visit our site and explore the collections. <br>"

					+ " <br>Our Online electronics is growing in a larger
amount these days and we are in high demand so we thanks all of you for "

					+ "making us up to that level. We Deliver Product to
your house with no extra delivery charges and we also have collection of most of the"

					+ "branded items. As a Welcome gift for our New
Customers we are providing additional 10% OFF Upto 500 Rs for the first product
purchase. To avail this offer you only have "

					+ "to enter the promo code given below.<br><br>
PROMO CODE: "
```

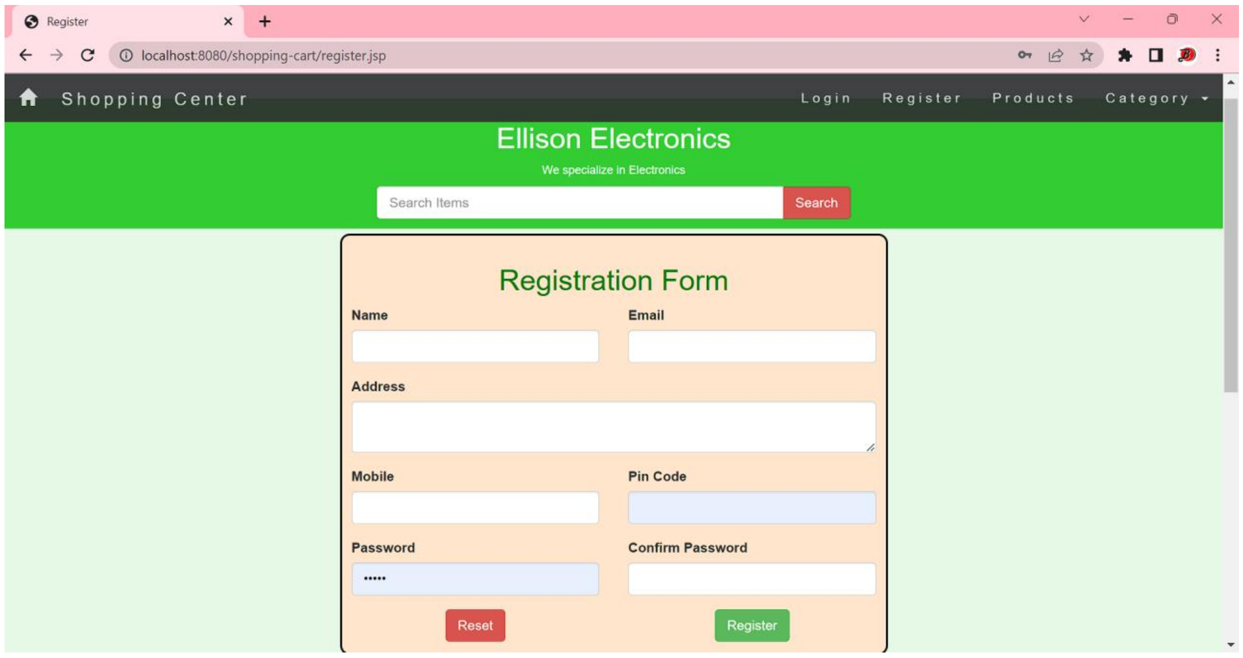
```
                                        + "<span
class='greenText'>ELLISON500</span><br><br>" + "Have a good day!<br>" + "</p>" +
"</body>"

                                        + "</html>";

                        JavaMailUtil.sendMail(recipient, subject, htmlTextMessage);

                        System.out.println("Mail Sent Successfully!");


                } catch (MessagingException e) {

                        System.out.println("Mail Sending Failed With Error: " +
e.getMessage());

                        e.printStackTrace();

                }

        }
```

V. RESULT AND DISCUSSION

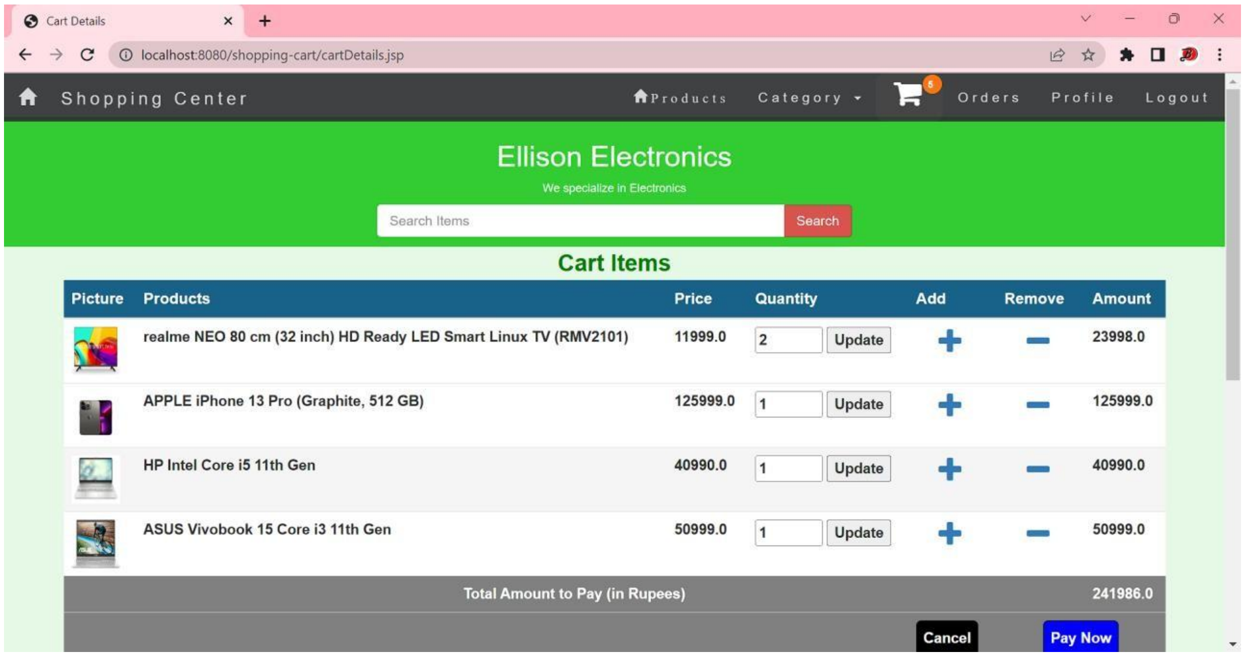
HOME PAGE:


Ellison Electronics
localhost:8080/shopping-cart/index.jsp
Shopping Center
Login
Register
Products
Category
Ellison Electronics
We specialize in Electronics
Search Items
Search
All Products
APPLE iPhone 13 Pro (Graphite, 512 GB)
iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures wit..
Rs 125999.0
Add to Cart
Buy Now
HP Intel Core i5 11th Gen
Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural..
Rs 40990.0
Add to Cart
Buy Now
LED Smart Google TV
Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos..
Rs 41999.0
Add to Cart
Buy Now

REGISTRATION PAGE:



CART ITEMS:



Cart Details

localhost:8080/shopping-cart/cartDetails.jsp

Shopping Center | Products | Category | Orders | Profile | Logout

Ellison Electronics

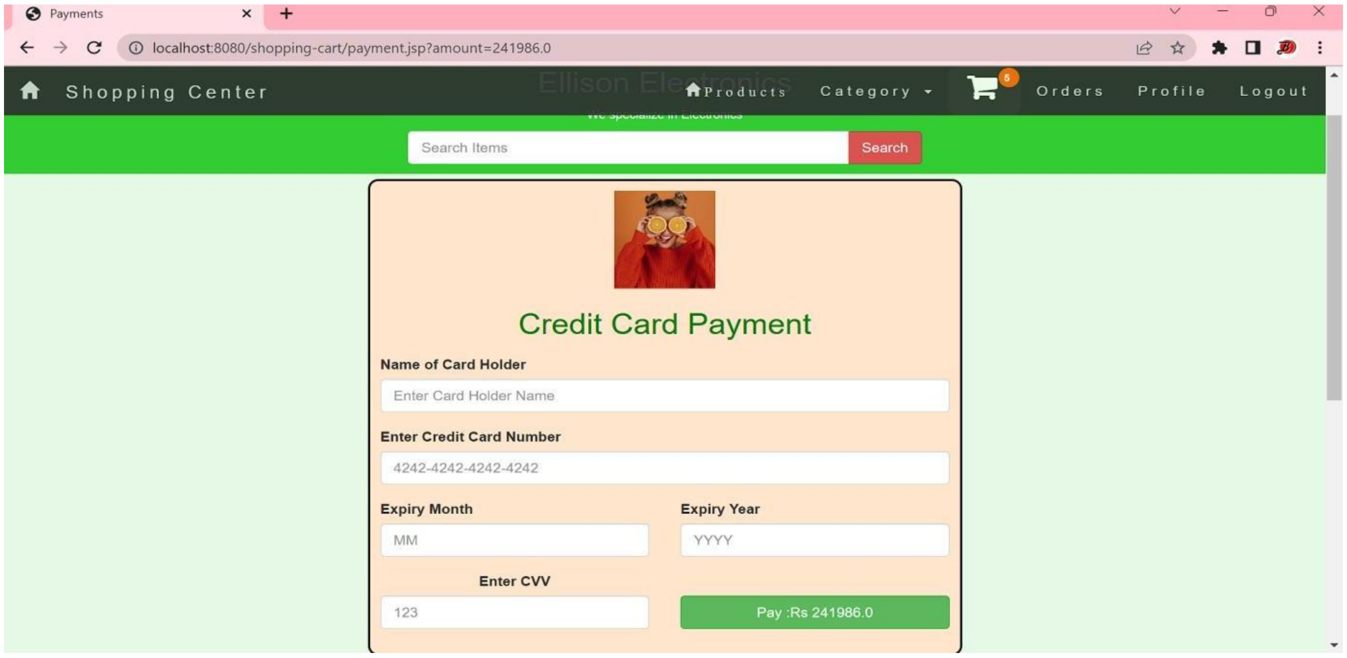
We specialize in Electronics

Cart Items

Picture	Products	Price	Quantity	Add	Remove	Amount
	realme NEO 80 cm (32 inch) HD Ready LED Smart Linux TV (RMV2101)	11999.0	2 Update	+	−	23998.0
	APPLE iPhone 13 Pro (Graphite, 512 GB)	125999.0	1 Update	+	−	125999.0
	HP Intel Core i5 11th Gen	40990.0	1 Update	+	−	40990.0
	ASUS Vivobook 15 Core i3 11th Gen	50999.0	1 Update	+	−	50999.0
	Total Amount to Pay (in Rupees)					241986.0

Cancel | Pay Now

CREDIT CARD PAYMENT:



Payments

localhost:8080/shopping-cart/payment.jsp?amount=241986.0

Shopping Center | Products | Category | Orders | Profile | Logout

Credit Card Payment

Name of Card Holder: Enter Card Holder Name

Enter Credit Card Number: 4242-4242-4242-4242

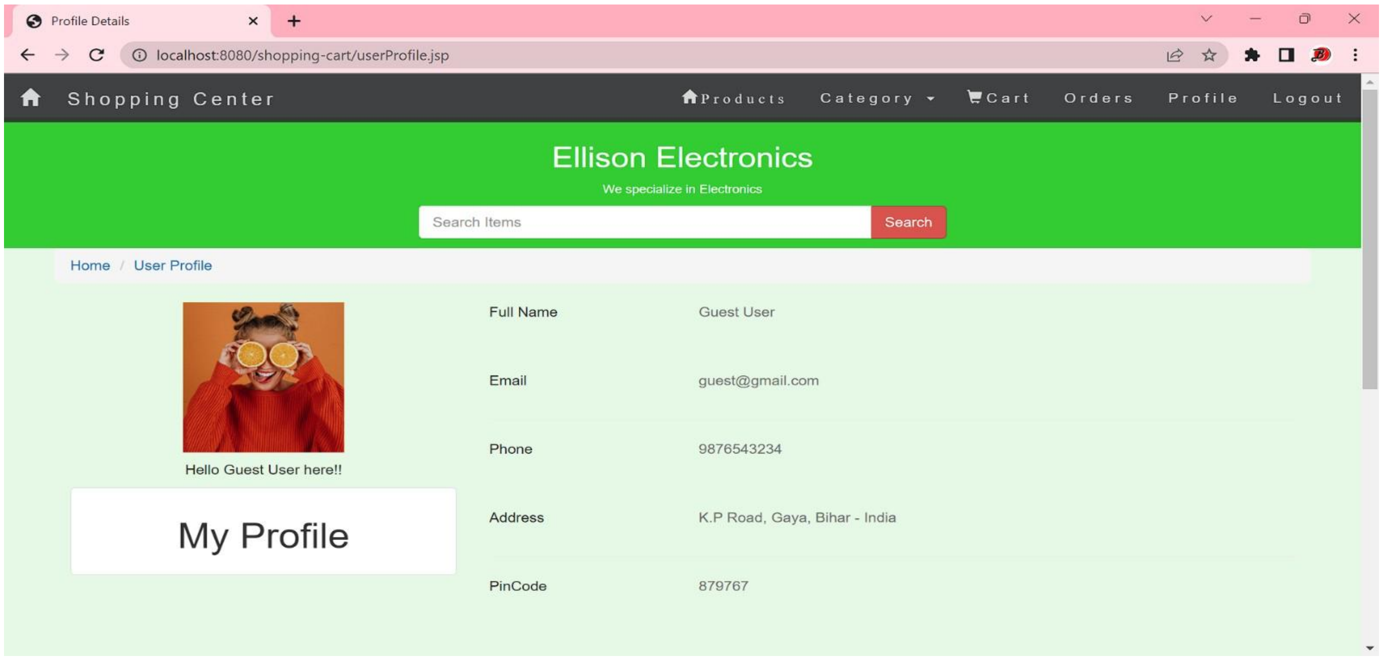
Expiry Month: MM

Expiry Year: YYYY

Enter CVV: 123

Pay :Rs 241986.0

USER PROFILE:


Profile Details

localhost:8080/shopping-cart/userProfile.jsp

Shopping Center | Products | Category | Cart | Orders | Profile | Logout

Ellison Electronics

We specialize in Electronics

Search Items | Search

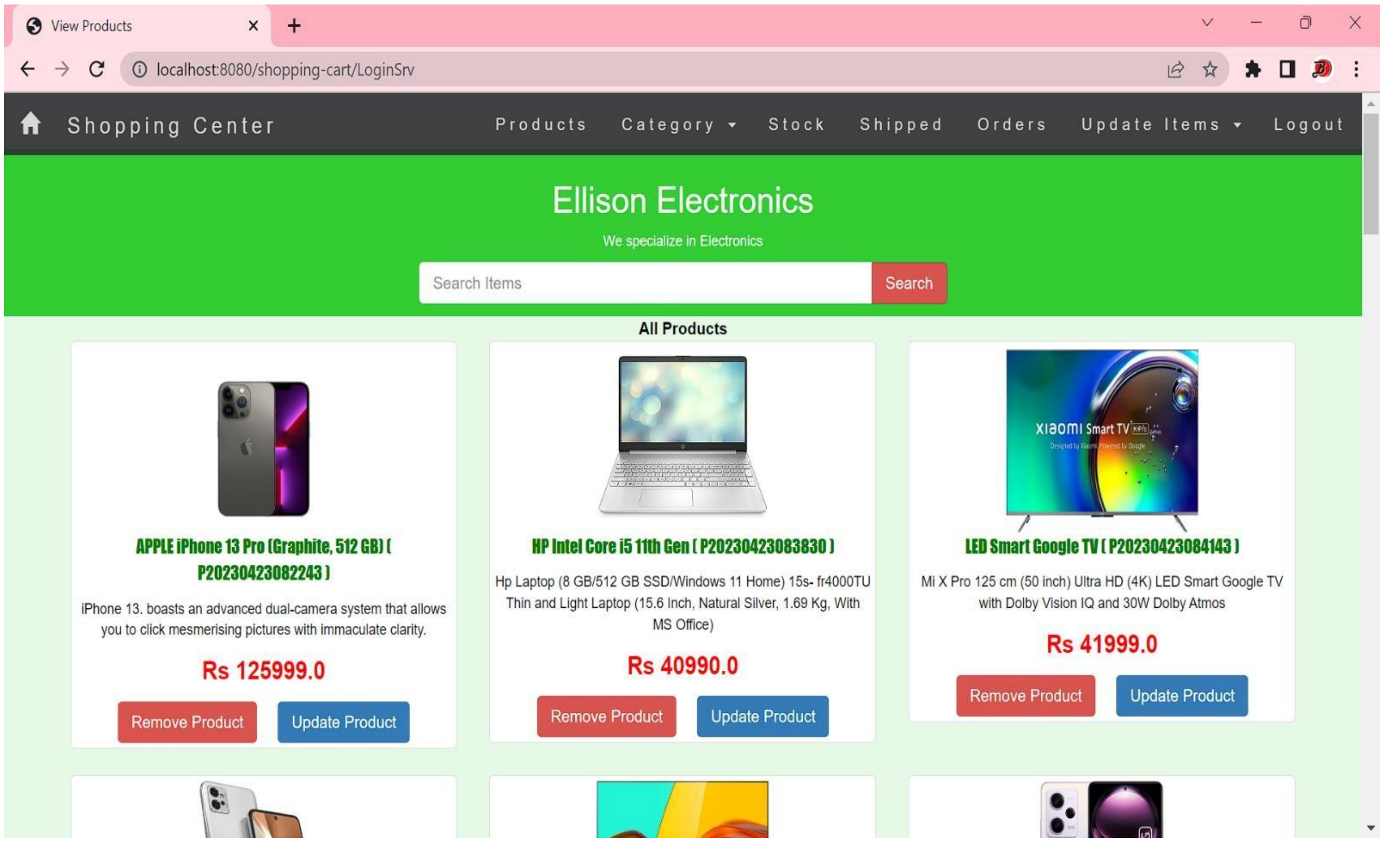
Home / User Profile

Hello Guest User here!!

My Profile

Full Name	Guest User
Email	guest@gmail.com
Phone	9876543234
Address	K.P Road, Gaya, Bihar - India
PinCode	879767

ADMIN HOME:


View Products

localhost:8080/shopping-cart/LoginSrv

Shopping Center | Products | Category | Stock | Shipped | Orders | Update Items | Logout

Ellison Electronics

We specialize in Electronics

Search Items | Search

All Products

APPLE iPhone 13 Pro (Graphite, 512 GB) [P20230423082243]

iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures with immaculate clarity.

Rs 125999.0

Remove Product | Update Product

HP Intel Core i5 11th Gen [P20230423083830]

Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural Silver, 1.69 Kg, With MS Office)

Rs 40990.0

Remove Product | Update Product

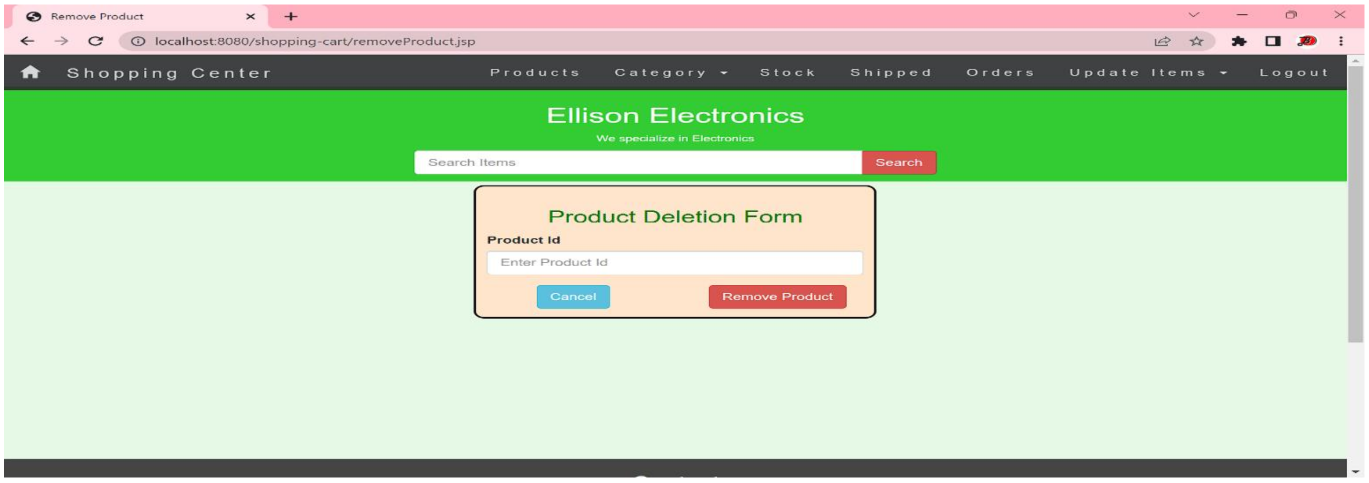
LED Smart Google TV [P20230423084143]

Mi X Pro 125 cm (50 inch) Ultra HD (4K) LED Smart Google TV with Dolby Vision IQ and 30W Dolby Atmos

Rs 41999.0

Remove Product | Update Product

REMOVE PRODUCT FROM STOCK:



Product Deletion Form

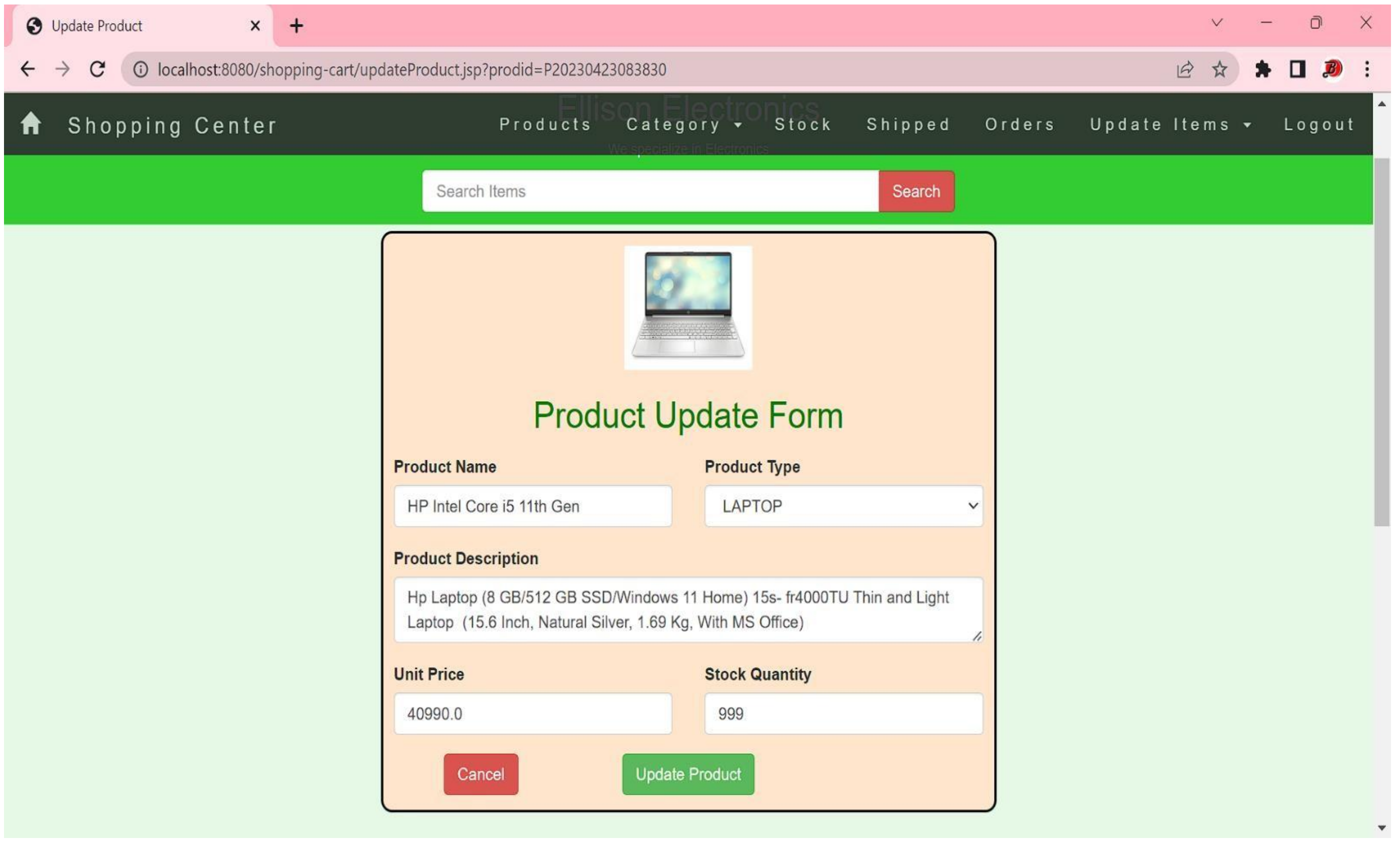
Product Id

Enter Product Id

Cancel

Remove Product

UPDATE STOCK ITEM:



Product Update Form

Product Name	Product Type
HP Intel Core i5 11th Gen	LAPTOP

Product Description

Hp Laptop (8 GB/512 GB SSD/Windows 11 Home) 15s- fr4000TU Thin and Light Laptop (15.6 Inch, Natural Silver, 1.69 Kg, With MS Office)

Unit Price	Stock Quantity
40990.0	999

Cancel

Update Product

RESULT

The online shopping cart web application developed using Java, JDBC, and Servlets is a comprehensive system designed to facilitate online shopping experiences. The key features and functionalities implemented in the application include:

1. Managing Products and Shopping Cart: Users can add, remove, and view products in their shopping cart.
2. Order Management: The system handles order placement, order status tracking, and order history.
3. Admin Functionality: Admin users have the ability to add, update, and remove products, manage user orders, and view sales data.
4. Separation of User and Admin Access: Clear distinctions are made between regular users and admin functionalities, ensuring appropriate access control.
5. User-Friendly Interface: The frontend is designed using HTML, CSS, JavaScript, and Bootstrap for a responsive and intuitive user experience.

The application utilizes a technology stack consisting of Java, JDBC, and Servlets for backend development, with MySQL as the relational database for storing product and user data. This technology stack ensures that the system is both efficient and scalable, providing users with a seamless shopping experience.

VI. CONCLUSION

The online shopping cart web application developed using Java, JDBC, and Servlets is a robust, feature-rich system that meets the core requirements of an e-commerce platform. The application offers a reliable and user-friendly experience for managing products, handling shopping cart functionality, processing orders, and ensuring appropriate admin-user separation.

The key strengths of this mini-project include:

1. Comprehensive Functionality: The application covers essential features like product management, user registration and authentication, shopping cart handling, order processing, and admin controls. This wide range of functionalities demonstrates the developer’s thorough understanding of e-commerce operations and their ability to deliver a complete, working solution.
2. Separation of Concerns: The clear demarcation between admin and user functionalities, with appropriate access controls, ensures that the system is secure and well-organized. This design allows for easier maintenance and scalability as the business evolves, and ensures that administrative tasks are performed securely while preventing unauthorized access.
3. Robust Technology Stack: The choice of Java, JDBC, and Servlets for the backend development, combined with HTML, CSS, JavaScript, and Bootstrap for the frontend, creates a solid and scalable

foundation for the application. The use of this technology stack allows for efficient server-side processing, secure database interactions, and a visually appealing user interface, ensuring both functionality and user experience.

4. Potential for Expansion: While the current version of the application meets the initial requirements, there are numerous opportunities for further enhancements. These include integrating advanced authentication mechanisms, payment gateways for secure transactions, and adding search and filtering options to enhance product discovery. Additionally, a recommendation system based on user preferences and purchase history could further personalize the user experience, increasing customer satisfaction and engagement.

In conclusion, the online shopping cart web application is a well-implemented mini-project that highlights the developer's skills in Java, JDBC, Servlets, and web development principles. The comprehensive features, secure design, and solid technology stack make the application a promising foundation for further development. With continued improvements and innovative features, this project has strong potential to evolve into a successful and competitive e-commerce platform.

VII. REFERENCES

- Design and Implementation of an Online Shopping Cart System **here**
- A Review on Online Shopping Cart Systems **here**
- E-commerce: Online Shopping Cart and Inventory Management System here
- Online Shopping Cart System and Its Impact on Consumer Behavior here
- Challenges and Solutions in Online Shopping Cart Systems here