E-COMMERCE PLATFORM

A PROJECT REPORT

Submitted by

SOFIYA C (2303811724322106)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"E-COMMERCE PLATFORM"** is the bonafide work of **SOFIYA C (2303811724322106)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.

S. Geetha

Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24

S. Geetha

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**E- COMMERCE PLATFORM**” is the result of original work done by us and best of our knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**

C. Sof

Signature

SOFIYA C

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”,** for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

E-commerce platforms have revolutionized the way businesses and consumers interact in the digital age, becoming a cornerstone of global trade and retail. These platforms serve as virtual marketplaces that facilitate the buying and selling of goods and services over the internet. Powered by advancements in technology, e-commerce platforms provide a seamless shopping experience, offering extensive product catalogs, secure payment gateways, and personalized recommendations driven by data analytics and artificial intelligence.

This paper explores the design, functionality, and impact of e-commerce platforms in various domains, including retail, business-to-business (B2B) transactions, and service industries. Modern e-commerce platforms combine scalable cloud infrastructure, responsive user interfaces, and advanced backend systems to manage vast volumes of users and transactions. Key features such as mobile responsiveness, integrated marketing tools, and real-time inventory management enhance user engagement and operational efficiency.

Additionally, the integration of digital payment systems, including wallets and cryptocurrencies, has streamlined the checkout process, making online shopping more convenient than ever.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

E-commerce platforms have become a transformative force in the global marketplace, enabling businesses and consumers to interact seamlessly in a digital environment. These platforms serve as online ecosystems that facilitate the buying and selling of products and services, transcending geographical and temporal barriers.

Global Accessibility and Convenience: E-commerce platforms enable businesses to reach a global audience and operate 24/7, breaking geographical and time constraints.

Technological Integration and Innovation: Advanced technologies like artificial intelligence, machine learning, and cloud computing drive e-commerce platforms.

Challenges and Future Prospects: Despite their advantages, e-commerce platforms face challenges such as cybersecurity threats, intense competition, and sustainability concerns**.**

1.2 OBJECTIVE

The primary objective of an E-commerce Platform is to create a seamless digital marketplace where buyers and sellers can interact, conduct transactions, and build long-term relationships. By leveraging advanced technologies, these platform aim to simplify the buying process, improve operational efficiency, and expand market reach for businesses of all sizes. Below are the detailed objectives of an e- commerce platform:

Facilitating Global Reach and Accessibility

· Enable businesses to connect with a vast audience regardless of geographical boundaries.

· Provide customers with access to a wide range of products and services 24/7**.**

Enhancing User Experience

· Deliver a user-friendly interface for easy navigation, product discovery, and transaction processing.

· Incorporate features such as personalized recommendations and advanced search filters to improve customer satisfaction.

Streamlining Transactions and Payments

- Simplify the Payment Process: E-commerce platforms integrate secure and efficient payment gateways to ensure smooth, hassle-free transactions. Offering multiple payment options such as credit cards, digital wallets, and bank transfers enhances customer convenience and encourages repeat business.
- Ensure Transaction Security: Implementing robust security measures like SSL encryption, fraud detection systems, and secure payment gateways ensures the protection of sensitive customer data, fostering trust between buyers and sellers.

Supporting Business Growth and Scalability

- Enable Business Expansion: By providing tools for inventory management, order tracking, and customer relationship management (CRM), e-commerce platforms help businesses scale quickly and manage growth.
- Increase Market Reach: In addition to global reach, e-commerce platforms help businesses tap into niche markets, target specific customer segments, and adjust their offerings based on market demand, further enhancing their growth potential.

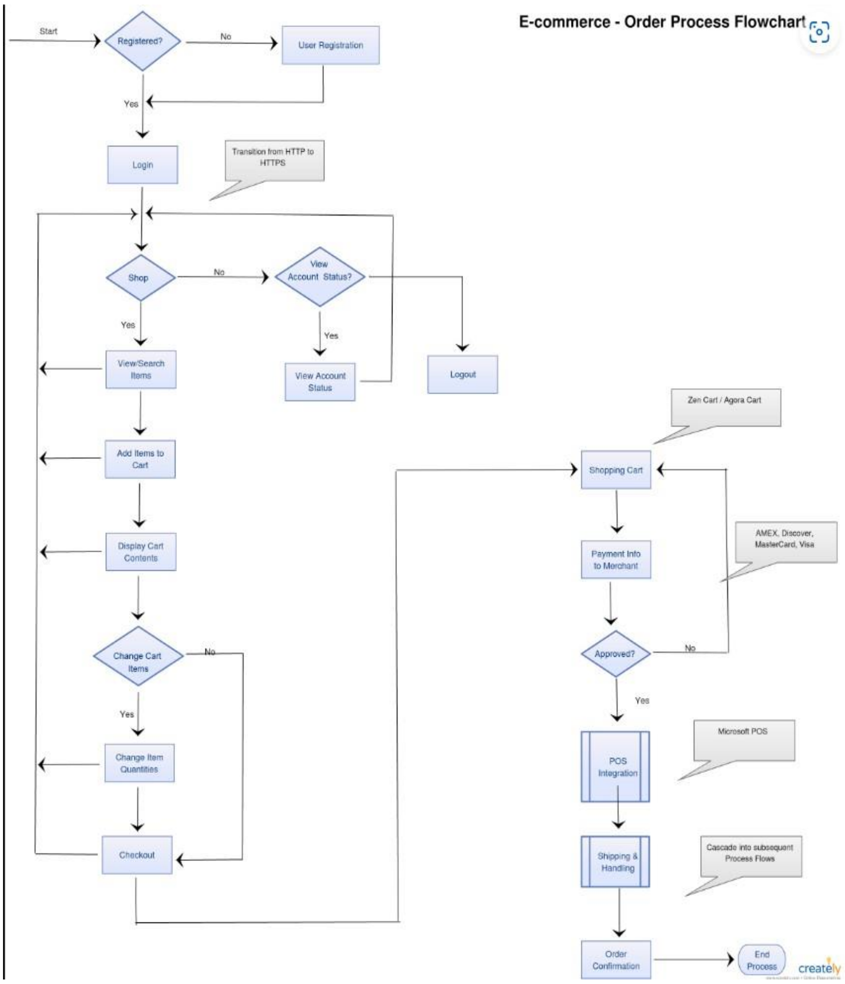
CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work aims to develop a comprehensive and efficient e-commerce platform that addresses modern consumer and business needs while integrating advanced technologies. The platform will focus on delivering a seamless user experience, robust backend functionality, and secure operations. Below are the key components of the proposed work:

1. User Management
2. Product Management
3. Order and Payment Processing
4. Recommendation and Search
5. Analytics and Reporting

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 CLASS OVERVIEW

- User: Represents the users of the application. Each user has a username and a password.
- Admin: Similar to User but for admins. Admins manage products and the catalog.
- Product: Represents products in the catalog, including the productName and price.
- MainMenu: The entry point to the application, providing options to login as a user or admin or register a new user.
- UserLogin: A GUI for users to log in.
- UserRegistration: A GUI for new users to register.
- AdminLogin: A GUI for admins to log in.
- AdminMenu: A menu for admins to add products or view the product catalog.
- AddProduct: A GUI for admins to add new products to the catalog.
- UserMenu: A menu for logged-in users, allowing them to search products, view their cart, or check out.

Swing Components

- JFrame: Represents a window.
- JPanel: A container for organizing components.
- JButton: Buttons that perform actions when clicked.
- JTextField: Text fields for user input.
- JPasswordField: Text field for securely inputting passwords.
- JLabel: A label to display text.
- JOptionPane: Used to display pop-up messages for alerts, confirmations, etc.
- GridLayout: Layout manager to arrange components in a grid.

Functional Flow

- Main Menu: Displays options to log in as a user, register a user, or log in as an admin. The main menu also provides an option to exit the application.
- User Registration: Allows a new user to create an account by entering a username and password. Upon successful registration, the user is returned to the main menu.
- User Login: Users log in with their credentials. If the login is successful, the UserMenu is shown.
- Admin Login: Admins can log in with predefined credentials to manage products.
- Admin Menu: After login, the admin can add products or view existing products.
- Product Management: Admins can add new products to the catalog using the AddProduct frame.
- User Menu: Logged-in users can:
    - Search products by name.
    - Add products to their cart.
    - View their cart.
    - Proceed to checkout, where they can choose between online or offline payment methods.

3.2DATA STRUCTURE:

- ArrayList:
    - users: Stores the list of registered users.
    - admins: Stores the list of admin users.
    - cart: Holds the products added to the cart by the logged-in user.
- HashMap:
    - productCatalog: Maps product names to Product objects. This serves as the catalog of available products.

Key Functionality

- User Login: The program checks if the entered username and password match any registered user. If successful, the user is redirected to their menu (UserMenu).
- Product Search: Users can search for products by name. If the product exists in the catalog, it can be added to the user's cart.
- Cart Management: Users can view the products in their cart, along with the total price, and proceed to checkout.
- Checkout Process: Users are prompted to select a payment method (Online or Offline). The program simulates a successful payment and clears the cart after purchase.
- Admin Functions: Admins can add products to the catalog and view all available products.

Swing Event Handling

- ActionListener is used to handle button clicks, where the program responds by performing the associated action (e.g., login, adding a product, viewing the cart).

Error Handling

- Basic error handling is implemented when:
    - Users enter incorrect credentials during login.
    - A user tries to add a product with an invalid price format.
    - A user attempts to check out without any products in the cart.

Enhancements for Future:

- Persistence: Right now, the data (users, admins, products) is stored in memory. To make the application more realistic, you could implement data persistence (e.g., using a database or file storage).
- Security: Passwords are stored as plain text. In a real-world application, passwords should be hashed and salted.
- Search Enhancements: Implement more advanced search features (e.g., filtering by category, price range).

CHAPTER 4
MODULE DESCRIPTION

4.1 USER MANAGEMENT MODULE:

· Purpose: This module handles all user-related functionalities, ensuring a seamless experience for both customers and administrators

· Operations**:**

User Registration and Authentication

Role-Based Access Control (RBAC)

4.2 PRODUCT MODULE:

· Purpose**:** This module enables sellers or administrators to add, update, delete, and manage product information. It organizes products into categories, subcategories, and tags for easy navigation.

· Operations:

Adding Products

Categorization

Inventory Management

4.3ORDER AND PAYMENT PROCESSING MODULE:

Purpose: This module manages the end-to-end process of placing orders, processing payments, and ensuring successful transactions.

Operations:

Shopping Cart

Order Checkout

4.4RECOMMENDATION AND SEARCH MODULE

Recommendation Module

· Purpose: This module uses search algorithms and AI-driven recommendation systems to enhance product discovery.

· Operations:

User Behavior Tracking

Collaborative Filtering

Search Module

· Purpose: The modules of an e-commerce platform are critical for ensuring seamless operation and enhancing user experience. Here's an overview of key modules typically found in such platforms

· Operations:

Search Bar and Autocomplete

Filters and Sorting

4.5 ANALYTICS AND REPORTING

Analytics Module

· Purpose: This module provides insights into platform performance, user behavior, and sales trends.

· Operations:

Sales and Performance Analysis

Customer Behavior Insights

Reporting Module

·Purpose: The Reporting Module in an e-commerce platform is crucial for providing insights into business performance, customer behavior, and operational efficiency

· Operations: Sales and Financial Reporting

CHAPTER 5

CONCLUSION

In conclusion, an e-commerce platform significantly transforms the way businesses operate and engage with customers. By leveraging the diverse features such as personalized shopping experiences, advanced search functionalities, and streamlined payment processing, businesses can enhance customer satisfaction and increase conversion rates. The automation of inventory management and order processing reduces operational inefficiencies, ensuring smoother transactions and quicker fulfillment, which in turn boosts customer trust and loyalty.

The data-driven insights offered by the platform provide valuable feedback that informs business strategies, from marketing campaigns to product development. These analytics empower businesses to tailor their offerings, optimize inventory, and make more accurate forecasts. Additionally, the scalability of the platform ensures that as the business grows, the system can adapt to increasing demands, helping to future-proof the business

REFERENCES:

Books:

"Java: A Beginner's Guide" by Herbert Schildt – This book covers Java basics and advanced topics, including GUI development with AWT/Swing. It’s ideal for understanding how to design interactive applications like your College Management System.

Websites:

Oracle Java Documentation – The official source for Java documentation, including AWT, Swing, and other Java libraries used in your project.

GeeksforGeeks - Java AWT & Swing – Offers detailed tutorials and code examples for working with AWT and Swing to create interactive user interfaces.

APPENDICES

APPENDIX A – SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.HashMap;

public class ShoppingApp {

    // Static lists to store users, admins, and product catalog
    private static ArrayList<User> users = new ArrayList<>();
    private static ArrayList<Admin> admins = new ArrayList<>();
    private static HashMap<String, Product> productCatalog = new HashMap<>();
    private static User loggedInUser = null;
    private static ArrayList<Product> cart = new ArrayList<>();  // User's cart

    // Main method to run the application
    public static void main(String[] args) {
        // Initialize sample data
        admins.add(new Admin("admin", "admin123"));
        productCatalog.put("Laptop", new Product("Laptop", 90000.00));
        productCatalog.put("Smartphone", new Product("Smartphone", 6000.00));
        productCatalog.put("Headphones", new Product("Headphones", 250.00));

        SwingUtilities.invokeLater(() -> new MainMenu());
```

```
}

// User Class
static class User {
    String username;
    String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }
}

// Admin Class
static class Admin {
    String username;
    String password;

    public Admin(String username, String password) {
        this.username = username;
        this.password = password;
    }
}

// Product Class
static class Product {
    String productName;
    double price;
    public Product(String productName, double price) {
        this.productName = productName;
        this.price = price;
    }

    public String getProductName() {
        return productName;
    }

    public double getPrice() {
        return price;
    }
}

// MainMenu Frame (Starting point)
static class MainMenu extends JFrame {
    public MainMenu() {
        setTitle("Shopping App");
        setSize(300, 200);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 1));

        JButton userLoginButton = new JButton("User Login");
        JButton userRegisterButton = new JButton("User Registration");
        JButton adminLoginButton = new JButton("Admin Login");
        JButton exitButton = new JButton("Exit");
}

        userLoginButton.addActionListener(e -> {
            dispose();
            new UserLogin();
        });

        userRegisterButton.addActionListener(e -> {
            dispose();
            new UserRegistration();
        });

        adminLoginButton.addActionListener(e -> {
            dispose();
            new AdminLogin();
        });

        exitButton.addActionListener(e -> System.exit(0));

        panel.add(userLoginButton);
        panel.add(userRegisterButton);
        panel.add(adminLoginButton);
        panel.add(exitButton);

        add(panel);
        setVisible(true);
    }
}

// User Login Frame
```

```
    static class UserLogin extends JFrame {
        public UserLogin() {
            setTitle("User Login");
            setSize(300, 200);
            setDefaultCloseOperation(DISPOSE_ON_CLOSE);
            setLocationRelativeTo(null);

            JPanel panel = new JPanel(new GridLayout(3, 2));
            JLabel usernameLabel = new JLabel("Username:");
            JTextField usernameField = new JTextField();
            JLabel passwordLabel = new JLabel("Password:");
            JPasswordField passwordField = new JPasswordField();
            JButton loginButton = new JButton("Login");
            JButton backButton = new JButton("Back");

            loginButton.addActionListener(e -> {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                for (User user : users) {
                    if (user.username.equals(username) &&
user.password.equals(password)) {
                        loggedInUser = user;
                        JOptionPane.showMessageDialog(this, "Login successful. Welcome
" + username + "!");
                        dispose();
                        new UserMenu();
                        return;
                    }
                }
            JOptionPane.showMessageDialog(this, "Invalid username or password.");
        });

        backButton.addActionListener(e -> {
            dispose();
            new MainMenu();
        });

        panel.add(usernameLabel);
        panel.add(usernameField);
        panel.add(passwordLabel);
        panel.add(passwordField);
        panel.add(loginButton);
        panel.add(backButton);

        add(panel);
        setVisible(true);
    }
}

// User Registration Frame
static class UserRegistration extends JFrame {
    public UserRegistration() {
        setTitle("User Registration");
        setSize(300, 200);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new GridLayout(3, 2));
```

```
JLabel usernameLabel = new JLabel("Username:");
JTextField usernameField = new JTextField();
JLabel passwordLabel = new JLabel("Password:");
JPasswordField passwordField = new JPasswordField();
JButton registerButton = new JButton("Register");
JButton backButton = new JButton("Back");

registerButton.addActionListener(e -> {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());
    users.add(new User(username, password));
    JOptionPane.showMessageDialog(this, "Registration successful.");
    dispose();
    new MainMenu();
});

backButton.addActionListener(e -> {
    dispose();
    new MainMenu();
});

panel.add(usernameLabel);
panel.add(usernameField);
panel.add(passwordLabel);
panel.add(passwordField);
panel.add(registerButton);
panel.add(backButton);

add(panel);
```

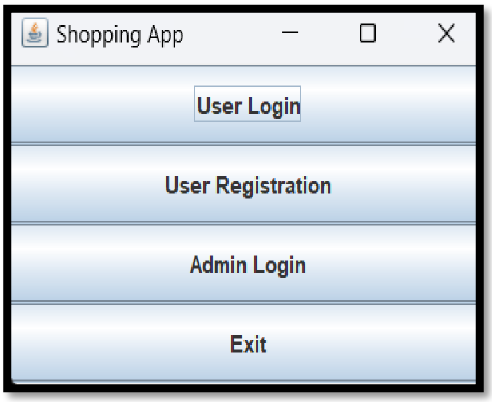
```
            setVisible(true);
        }
    }

    // Admin Login Frame
    static class AdminLogin extends JFrame {
        public AdminLogin() {
            setTitle("Admin Login");
            setSize(300, 200);
            setDefaultCloseOperation(DISPOSE_ON_CLOSE);
            setLocationRelativeTo(null);

            JPanel panel = new JPanel(new GridLayout(3, 2));
            JLabel usernameLabel = new JLabel("Username:");
            JTextField usernameField = new JTextField();
            JLabel passwordLabel = new JLabel("Password:");
            JPasswordField passwordField = new JPasswordField();
            JButton loginButton = new JButton("Login");
            JButton backButton = new JButton("Back");

            loginButton.addActionListener(e -> {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
                for (Admin admin : admins) {
                    if (admin.username.equals(username) &&
admin.password.equals(password)) {
                        JOptionPane.showMessageDialog(this, "Admin login successful.");
                        dispose();
                        new AdminMenu();
                        return;
                    }
                }
                JOptionPane.showMessageDialog(this, "Invalid admin username or password.");
            });

            backButton.addActionListener(e -> {
                dispose();
                new MainMenu();
            });

            panel.add(usernameLabel);
            panel.add(usernameField);
            panel.add(passwordLabel);
            panel.add(passwordField);
            panel.add(loginButton);
            panel.add(backButton);

            add(panel);
            setVisible(true);
        }
    }

    // Admin Menu Frame
    static class AdminMenu extends JFrame {
        public AdminMenu() {
            setTitle("Admin Menu");
            setSize(300, 200);
```

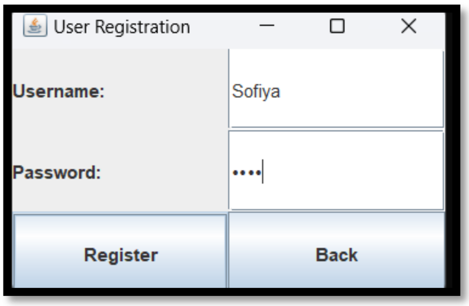
```
setDefaultCloseOperation(DISPOSE_ON_CLOSE);
setLocationRelativeTo(null);

JPanel panel = new JPanel(new GridLayout(3, 1));
JButton addProductButton = new JButton("Add Product");
JButton viewProductsButton = new JButton("View Products");
JButton logoutButton = new JButton("Logout");

addProductButton.addActionListener(e -> {
    dispose();
    new AddProduct();
});

viewProductsButton.addActionListener(e -> {
    StringBuilder products = new StringBuilder("Available Products:\n");
    for (String productName : productCatalog.keySet()) {
        Product product = productCatalog.get(productName);
        products.append("- ").append(product.getProductName()).append(": ₹").append(product.getPrice()).append("\n");
    }
    JOptionPane.showMessageDialog(this, products.toString());
});

logoutButton.addActionListener(e -> {
    dispose();
    new MainMenu();
});

panel.add(addProductButton);
```

```
        panel.add(viewProductsButton);
        panel.add(logoutButton);

        add(panel);
        setVisible(true);
    }
}

// Add Product Frame (Admin can add products)
static class AddProduct extends JFrame {
    public AddProduct() {
        setTitle("Add Product");
        setSize(300, 200);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new GridLayout(3, 2));
        JLabel productNameLabel = new JLabel("Product Name:");
        JTextField productNameField = new JTextField();
        JLabel priceLabel = new JLabel("Price:");
        JTextField priceField = new JTextField();
        JButton addButton = new JButton("Add");
        JButton backButton = new JButton("Back");

        addButton.addActionListener(e -> {
            String productName = productNameField.getText();
            try {
                double price = Double.parseDouble(priceField.getText());
                productCatalog.put(productName, new Product(productName, price));
                JOptionPane.showMessageDialog(this, "Product added successfully.");
                dispose();
                new AdminMenu();
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(this, "Invalid price.");
            }
        });

        backButton.addActionListener(e -> {
            dispose();
            new AdminMenu();
        });

        panel.add(productNameLabel);
        panel.add(productNameField);
        panel.add(priceLabel);
        panel.add(priceField);
        panel.add(addButton);
        panel.add(backButton);

        add(panel);
        setVisible(true);
    }
}

// User Menu Frame (Logged-in User)
static class UserMenu extends JFrame {
    public UserMenu() {
        setTitle("User Menu");
        setSize(300, 200);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        JPanel panel = new JPanel(new GridLayout(5, 1));
        JButton searchProductButton = new JButton("Search Product");
        JButton viewCartButton = new JButton("View Cart");
        JButton checkoutButton = new JButton("Proceed to Checkout");
        JButton logoutButton = new JButton("Logout");

        searchProductButton.addActionListener(e -> {
            String productName = JOptionPane.showInputDialog(this, "Enter product name:");
            Product product = productCatalog.get(productName);
            if (product != null) {
                JOptionPane.showMessageDialog(this, "Product found: " + product.getProductName() + " - ₹" + product.getPrice());
                int option = JOptionPane.showConfirmDialog(this, "Do you want to add it to your cart?");
                if (option == JOptionPane.YES_OPTION) {
                    cart.add(product);
                    JOptionPane.showMessageDialog(this, product.getProductName() + " added to your cart.");
                }
            } else {
                JOptionPane.showMessageDialog(this, "Product not found.");
            }
        });
        viewCartButton.addActionListener(e -> {
            if (cart.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Your cart is empty.");
            } else {
                StringBuilder cartDetails = new StringBuilder("Your Cart:\n");
                double total = 0;
                for (Product product : cart) {
                    cartDetails.append(product.getProductName()).append(":
₹").append(product.getPrice()).append("\n");
                    total += product.getPrice();
                }
                cartDetails.append("Total: ₹").append(total);
                JOptionPane.showMessageDialog(this, cartDetails.toString());
            }
        });

        checkoutButton.addActionListener(e -> {
            if (cart.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Your cart is empty.");
            } else {
                double total = 0;
                for (Product product : cart) {
                    total += product.getPrice();
                }

                // Payment Method Selection
                String paymentOption = JOptionPane.showInputDialog(this, "Choose
payment method: Online/Offline");
                if (paymentOption != null &&
paymentOption.equalsIgnoreCase("Online")) {
                    JOptionPane.showMessageDialog(this, "Processing online
payment...");
                    JOptionPane.showMessageDialog(this, "Payment successful! Total:
₹" + total);
                } else if (paymentOption != null &&
paymentOption.equalsIgnoreCase("Offline")) {
                    JOptionPane.showMessageDialog(this, "Offline payment selected.
Please visit the store.");
                } else {
                    JOptionPane.showMessageDialog(this, "Invalid payment method
selected.");
                    return;
                }

                cart.clear();  // Clear cart after payment
            }
        });

        logoutButton.addActionListener(e -> {
            dispose();
            new MainMenu();
        });

        panel.add(searchProductButton);
        panel.add(viewCartButton);
        panel.add(checkoutButton);
        panel.add(logoutButton);
```

```
            add(panel);
            setVisible(true);
        }
    }
}
```

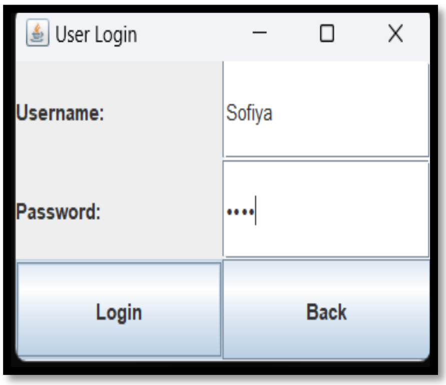
APPENDIX B – SCREENSHOTS





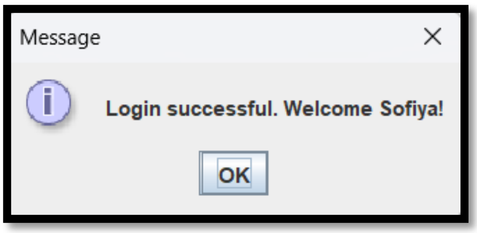
Username: Sofiya

Password: ••••

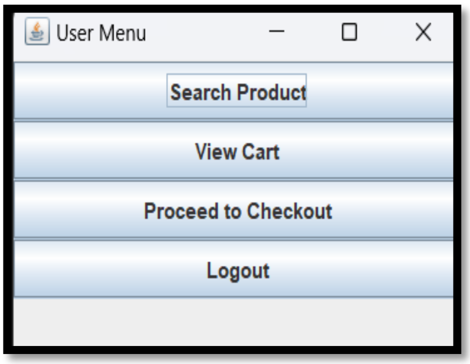


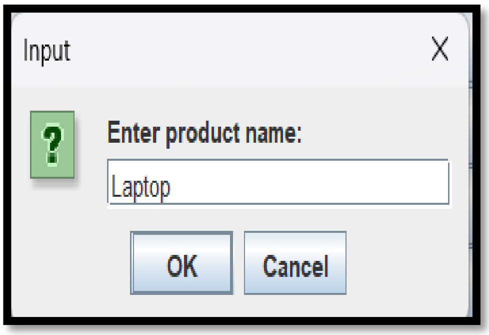
Username: Sofiya

Password: ••••

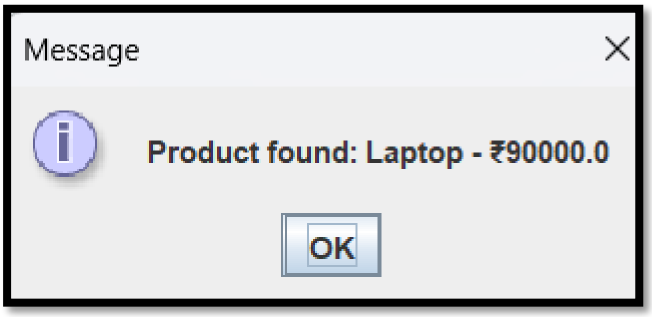


Login successful. Welcome Sofiya!

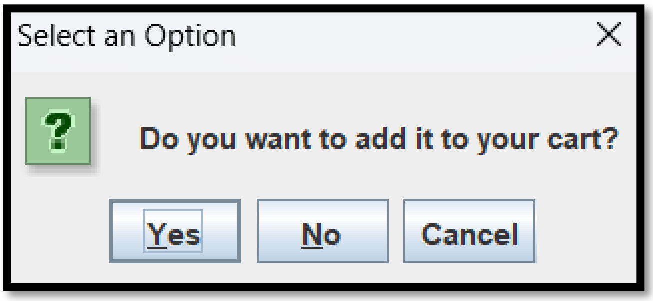




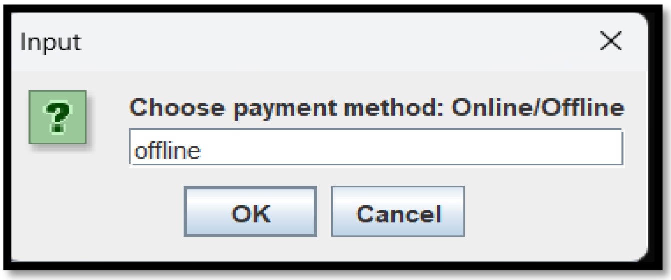
Enter product name: Laptop



Product found: Laptop - ₹90000.0

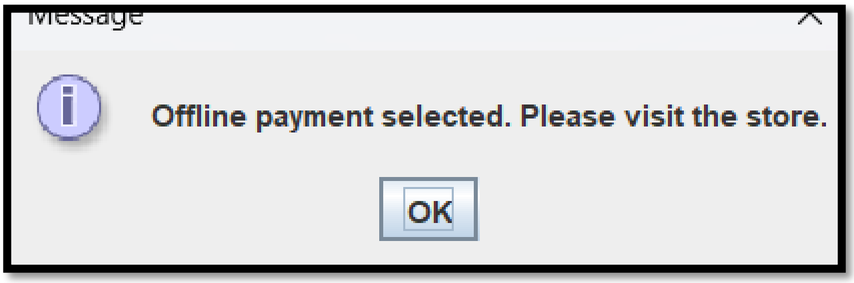


Do you want to add it to your cart?

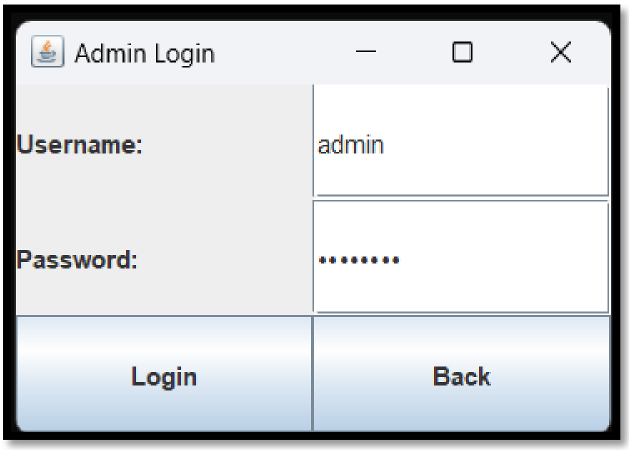


Choose payment method: Online/Offline

offline

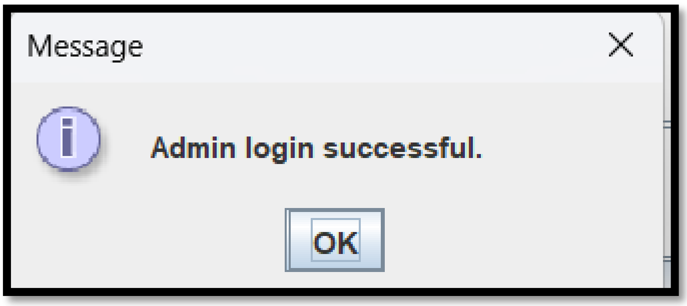


Offline payment selected. Please visit the store.

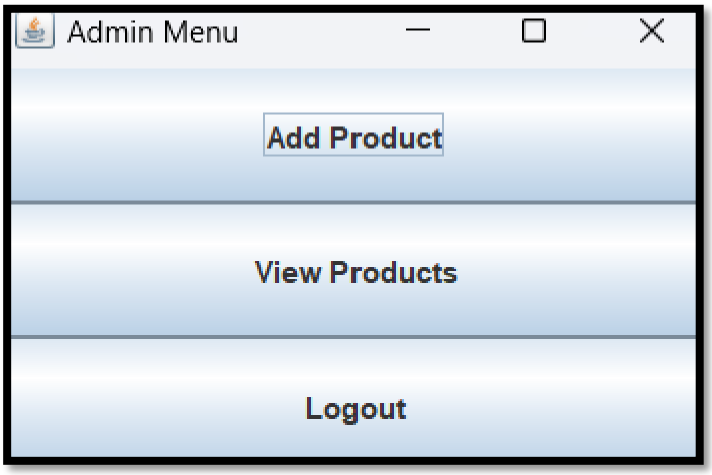


Username: admin

Password: ••••••••



Admin login successful.


Admin Menu
Add Product
View Products
Logout