SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be **University** *u/s 3 of UGC Act, 1956*

21CSC206P – Advanced Object-Oriented Programming

Project Review 2

E-Shopping Cart

1. Sneha Das – RA2311027010023
2. Dhanvi Chaudhary – RA2311027010031

Table of Contents

ABSTRACT

- The E-Shopping Mart project is an online shopping platform developed in Java, designed to provide a user-friendly interface for both customers and administrators. The system enables customers to browse products, manage their shopping carts and place orders.
- For administrators, the platform offers tools to manage product listings, categories and customer orders, ensuring efficient store management. Built using core Java principles and utilizing a relational database (such as MySQL) for data storage, the project implements Java Database Connectivity (JDBC) for seamless interaction between the application and the database.
- The system is developed as a desktop application using Java Swing or as a web-based platform with JSP/Servlets.
- This project demonstrates the practical application of Java for real-world e-commerce solutions and can be extended with additional features like real-time payment processing.

OBJECTIVE

- The primary objective of the E-Shopping Cart mini project is to develop a simple, user-friendly online shopping system using Java that enables customers to browse products, manage their shopping cart, and place orders efficiently.

- The system aims to provide basic e-commerce functionalities such as product selection, cart management, and order processing.

- By utilizing core Java concepts, JDBC for database connectivity and Java Swing for a desktop application, this project demonstrates the practical application of Java in building small-scale, real-world e-commerce solutions.

PROBLEM STATEMENT

- This Mini Project is based on E Shopping. We took up this topic because the offline system of shopping marts deals with a lot of inconveniences. This program prints the product, its details, add to cart option and the billing system. This project will make the shopping experience easier and faster with giving all the information about the product.

- Firstly, it (program) asks the users for login credentials if account exists or else asks the user to create a new account and saves the new account details in a SQL table named 'userid_pwd'. It then displays the Menu which consists the different categories of products such as Fresh Produce, Clothing, Home Appliances, Stationary, etc. Few of the categories are sub-categorised. The user can then choose their preferred category and select required products and add it to their cart which is then saved into the table named 'add_to_cart'. Further while checking out, it displays the final billing amount.

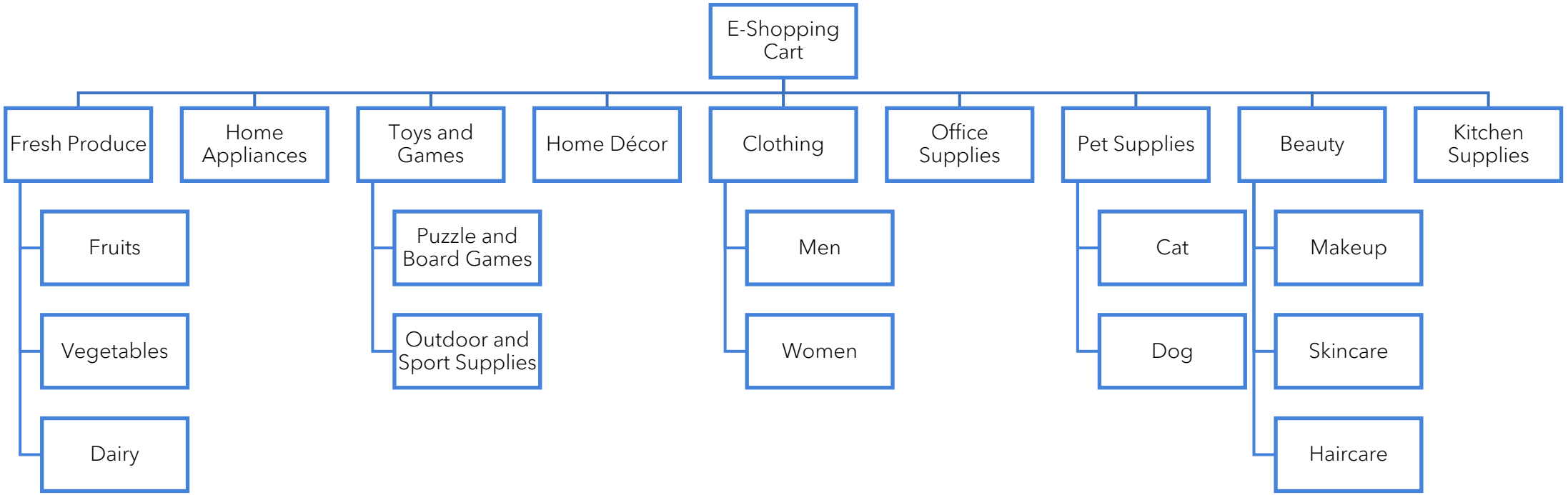
SOFTWARE REQUIREMENTS

- Programming Language – Java

- Java Libraries -
  - Java Swing – for the graphical user interface(GUI)
  - Java Database Connectivity(JDBC) – for connecting the java application to the MySQL database.

- Database Tool – MySQL Workbench for managing and interacting with the MySQL database.

DATABASE DESIGN



Proposed System

In this java swing based shopping cart, we use several logical modules according to the functionality of the code.

- User Authentication:

  In this module we authenticate the users by logging in and creating a new account. It interacts with MySQL table userid_pwd that stores user credentials: User_id and Pwd. It checks whether the user exists in the database or not and on account creation it inserts a new user derail into the database.

- Category Selection:

  After the user is logged in, they are presented with various product categories that are Fresh Produce, Toys and Games, Clothing, Pet Supplies, Beauty, Home décor, Home Appliance and Office Supplies. From these categories the user selects one to browse the available products.

- Sub-category Selection

  If a category contains a subcategories, the user is asked to choose a subcategory. This function fetches subcategory details from the database. Like from fruits, vegetables, dairy table under fresh produce.

- Item Selection:

  Once a category or subcategory is selected, this module displays the products available in that section with their prices.

  The user then selects the items to the add to their cart, which will be inserted to the table add_to_cart database.

- Database Interaction:

  This functionality is very important as it retrieves user and product data from MySQL Workbench.

  The system queries the database for categories, subcategories, and products. It also handles actions like adding products to the cart and storing new user details during sign-up.

To overcome the limitations of the existing methodology in the java Swing based e shopping cart application we can improve the functionality and design.

- Better code structure:

  The current system is very monolithic, with all functionality bundled in a single file. It makes the code hard to maintain, debug, and extend. We can write the code by breaking it into separate classes. A separate class can handle login and account creation. And a class can handle category, subcategory and products. One class can be for adding, viewing and editing cart items. Then one class can handle the overall GUI navigation.

- Improve Database Security:

  We can also improve the database security as now directly concatenating user input into SQL statements is a serious security risk. We can use use PreparedStatements for all database queries. This ensures user input is safely parameterized and prevents SQL injection attacks.

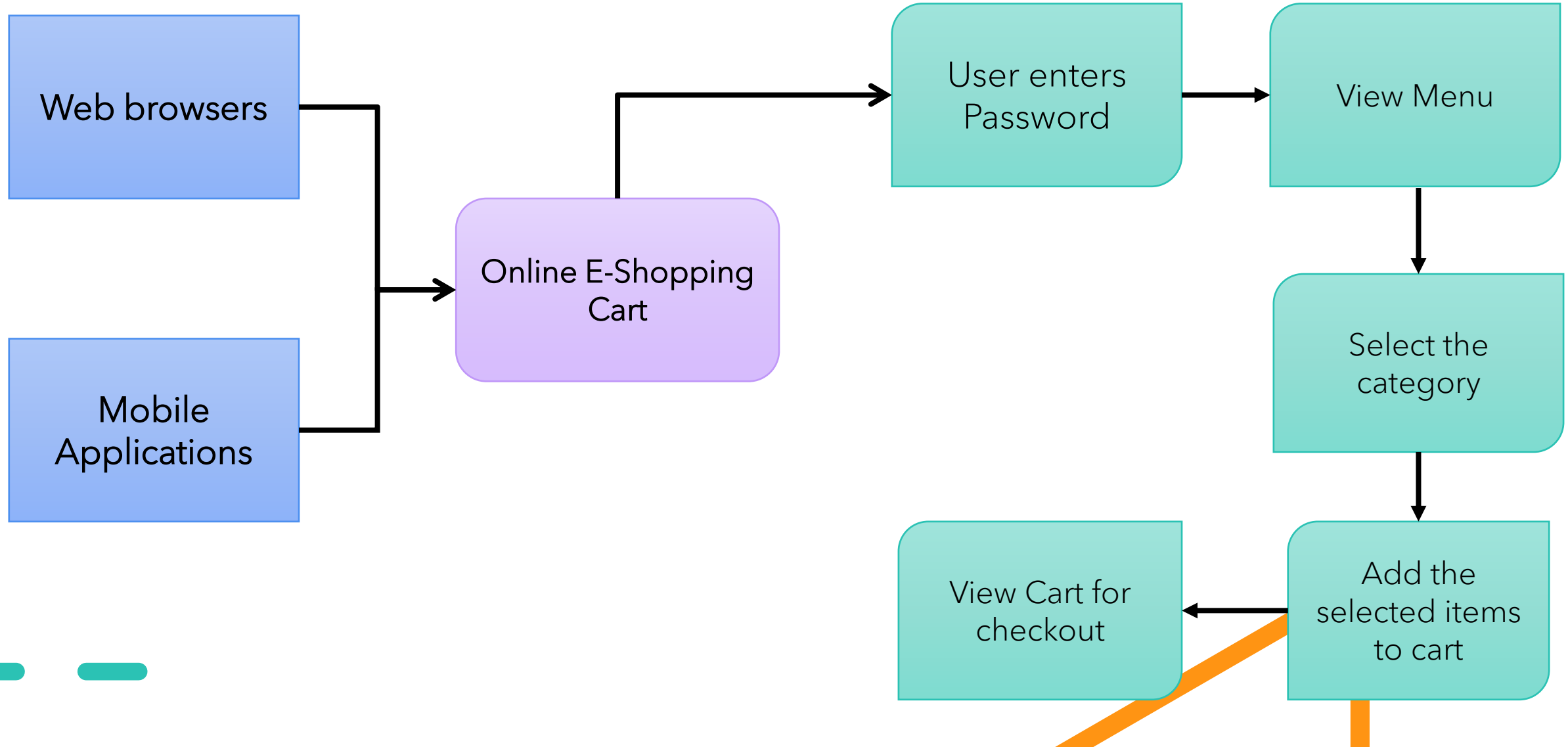
- Optimize Database Queries:

  The current system makes separate database queries for each action. This can be inefficient for large datasets. We can use batch queries or lazy loading to reduce database calls. For example, load categories and subcategories at once rather than making multiple queries. Optimize the SQL queries and ensure the database has proper indexes for faster data retrieval.

GUI DESIGN

- The **GUI (Graphical User Interface)** used for this mini project "**E-Shopping Cart**" is Java Swing. It provides a user-friendly interface for functionalities such as login, product listing, shopping cart and order confirmation.
- Java Swing is a GUI toolkit and a part of Java Foundation Classes (JFC) that is used to create window-based applications.
- It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.
- Unlike AWT, Java Swing provides platform-independent and lightweight components.
- The javax.swing package provides classes for java swing API such as JButton, JTextField,
- JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.
- We will be using Java Swing components like JFrame, JPanel, JTable, JButton, and JTextField.

ARCHITECTURE DIAGRAM



CODE SNIPPET

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```
6   class EShoppingCartGUI extends JFrame {
179     // Show Fresh Produce Subcategories
180     private void showFreshProduce() {
181         JPanel freshProducePanel = new JPanel(new GridLayout(rows:3, cols:1));
182         JButton fruitsButton = new JButton(text:"Fruits");
183         JButton vegetablesButton = new JButton(text:"Vegetables");
184         JButton dairyButton = new JButton(text:"Dairy");
185 
186         fruitsButton.addActionListener(e -> showItems(category:"fruits"));
187         vegetablesButton.addActionListener(e -> showItems(category:"vegetables"));
188         dairyButton.addActionListener(e -> showItems(category:"dairy"));
189 
190         freshProducePanel.add(fruitsButton);
191         freshProducePanel.add(vegetablesButton);
192         freshProducePanel.add(dairyButton);
193 
194         // Clear and switch to fresh produce panel
195         getContentPane().removeAll();
196         add(freshProducePanel);
197         revalidate();
198         repaint();
199     }
200     private void showToyandGames(){
201         JPanel toysandgamesPanel = new JPanel(new GridLayout(rows:2,cols:1));
202         JButton puzzlesButton = new JButton(text:"Puzzle and Board Games");
203         JButton outdoorsButton = new JButton(text:"Outdoors and Sports Toys");
204 
205         puzzlesButton.addActionListener(e -> addItemsToCart(category:"puzzle_and_board_games"));
206         outdoorsButton.addActionListener(e -> addItemsToCart(category:"outdoors_and_sports_toys"));
207 
208         toysandgamesPanel.add(puzzlesButton);
209         toysandgamesPanel.add(outdoorsButton);
210 
211         // Clear and switch to fresh produce panel
212         getContentPane().removeAll();
213         add(toysandgamesPanel);
214         revalidate();
```

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```
6   class EShoppingCartGUI extends JFrame {
118
119     // Create Account Function
120     private void createAccount() {
121         String username = createUsernameField.getText();
122         String password = new String(createPasswordField.getPassword());
123         String confirmPassword = new String(confirmPasswordField.getPassword());
124
125         if (!password.equals(confirmPassword)) {
126             messageLabel.setText(text:"Passwords do not match!");
127             return;
128         }
129
130         try {
131             String query = "INSERT INTO userid_pwd (User_id, Pwd) VALUES ('" + username + "', '" + password + "')";
132             stmt.executeUpdate(query);
133             messageLabel.setText(text:"Account Created Successfully!");
134         } catch (SQLException e) {
135             e.printStackTrace();
136         }
137     }
138
139     // Show Product Categories after login
140     private void showProductCategories() {
141         JPanel categoryPanel = new JPanel(new GridLayout(rows:8, cols:1));
142         JButton freshproduceButton = new JButton(text:"Fresh Produce");
143         JButton toysandgamesButton = new JButton(text:"Toys and Games");
144         JButton petsuppliesButton = new JButton(text:"Pet Supplies");
145         JButton clothingButton = new JButton(text:"Clothing");
146         JButton beautyButton = new JButton(text:"Beauty");
147         JButton homedecorButton = new JButton(text:"Home Decor");
148         JButton homeapplianceButton = new JButton(text:"Home Appliance");
149         JButton officesuppliesButton = new JButton(text:"Office Supplies");
150
151         freshproduceButton.addActionListener(e -> showFreshProduce());
152         toysandgamesButton.addActionListener(e -> showToyandGames());
153         petsuppliesButton.addActionListener(e -> showPetSupplies());
154         clothingButton.addActionListener(e -> showClothing());
```

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```
6   class EShoppingCartGUI extends JFrame {
140     private void showProductCategories() {
142         JButton freshproduceButton = new JButton(text:"Fresh Produce");
143         JButton toysandgamesButton = new JButton(text:"Toys and Games");
144         JButton petsuppliesButton = new JButton(text:"Pet Supplies");
145         JButton clothingButton = new JButton(text:"Clothing");
146         JButton beautyButton = new JButton(text:"Beauty");
147         JButton homedecorButton = new JButton(text:"Home Decor");
148         JButton homeapplianceButton = new JButton(text:"Home Appliance");
149         JButton officesuppliesButton = new JButton(text:"Office Supplies");
150
151         freshproduceButton.addActionListener(e -> showFreshProduce());
152         toysandgamesButton.addActionListener(e -> showToyandGames());
153         petsuppliesButton.addActionListener(e -> showPetSupplies());
154         clothingButton.addActionListener(e -> showClothing());
155         beautyButton.addActionListener(e -> showBeauty());
156         homedecorButton.addActionListener(e -> showItems(category:"home_decor"));
157         homeapplianceButton.addActionListener(e -> showItems(category:"home_appliance"));
158         officesuppliesButton.addActionListener(e -> showItems(category:"office_supplies"));
159
160         // Add action listeners for other categories...
161
162         categoryPanel.add(freshproduceButton);
163         categoryPanel.add(toysandgamesButton);
164         categoryPanel.add(petsuppliesButton);
165         categoryPanel.add(clothingButton);
166         categoryPanel.add(beautyButton);
167         categoryPanel.add(homedecorButton);
168         categoryPanel.add(homeapplianceButton);
169         categoryPanel.add(officesuppliesButton);
170
171
172         // Clear the frame and add the category panel
173         getContentPane().removeAll();
174         add(categoryPanel);
175         revalidate();
176         repaint();
177     }
```

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```java
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class EShoppingCartGUI extends JFrame {
    private Connection mycon;
    private Statement stmt;

    // GUI Components
    private JTextField usernameField;
    private JPasswordField passwordField;
    private JTextField createUsernameField;
    private JPasswordField createPasswordField;
    private JPasswordField confirmPasswordField;
    private JLabel messageLabel;

    public EShoppingCartGUI() {
        // Establish the database connection
        connectToDatabase();

        // Create the JFrame
        setTitle(title:"E-Shopping Cart");
        setSize(width:500, height:400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new CardLayout());

        // Create GUI for login and account creation
        initializeLoginAndAccountGUI();

        setVisible(b:true);
    }

    // Connect to MySQL Database
    private void connectToDatabase() {
        try {
            mycon = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/shoppingmart", user:"root", password:"sneha29");
            stmt = mycon.createStatement();
```

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```
6   class EShoppingCartGUI extends JFrame {
44      // Initialize GUI for Login and Account Creation
45      private void initializeLoginAndAccountGUI() {
46          JPanel mainPanel = new JPanel(new GridLayout(rows:3, cols:1));
47
48          // Login Panel
49          JPanel loginPanel = new JPanel();
50          loginPanel.setLayout(new GridLayout(rows:3, cols:2));
51
52          usernameField = new JTextField();
53          passwordField = new JPasswordField();
54          JButton loginButton = new JButton(text:"Login");
55
56          loginPanel.add(new JLabel(text:"Username:"));
57          loginPanel.add(usernameField);
58          loginPanel.add(new JLabel(text:"Password:"));
59          loginPanel.add(passwordField);
60          loginPanel.add(loginButton);
61
62          loginButton.addActionListener(e -> login());
63
64          // Account Creation Panel
65          JPanel accountPanel = new JPanel();
66          accountPanel.setLayout(new GridLayout(rows:4, cols:2));
67
68          createUsernameField = new JTextField();
69          createPasswordField = new JPasswordField();
70          confirmPasswordField = new JPasswordField();
71          JButton createAccountButton = new JButton(text:"Create Account");
72
73          accountPanel.add(new JLabel(text:"Create Username:"));
74          accountPanel.add(createUsernameField);
75          accountPanel.add(new JLabel(text:"Create Password:"));
76          accountPanel.add(createPasswordField);
77          accountPanel.add(new JLabel(text:"Confirm Password:"));
78          accountPanel.add(confirmPasswordField);
79          accountPanel.add(createAccountButton);
```

C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI

```
6   class EShoppingCartGUI extends JFrame {
45      private void initializeLoginAndAccountGUI() {
80
81          createAccountButton.addActionListener(e -> createAccount());
82
83          // Message Label for displaying errors
84          messageLabel = new JLabel(text:"", SwingConstants.CENTER);
85
86          mainPanel.add(loginPanel);
87          mainPanel.add(accountPanel);
88          mainPanel.add(messageLabel);
89
90          add(mainPanel);
91      }
92
93      // Login Function
94      private void login() {
95          String username = usernameField.getText();
96          String password = new String(passwordField.getPassword());
97
98          try {
99              String query = "SELECT * FROM userid_pwd WHERE User_id = ? AND Pwd = ?";
100             PreparedStatement pstmt = mycon.prepareStatement(query);
101             pstmt.setString(parameterIndex:1, username);
102             pstmt.setString(parameterIndex:2, password);
103
104             // Execute the query
105             ResultSet rs = pstmt.executeQuery();
106
107             if (rs.next()) {
108                 messageLabel.setText(text:"Login Successful");
109                 showProductCategories();  // Navigate to product categories
110             } else {
111                 messageLabel.setText(text:"Invalid Credentials");
112             }
113         } catch (SQLException e) {
114             e.printStackTrace();
115         }
```

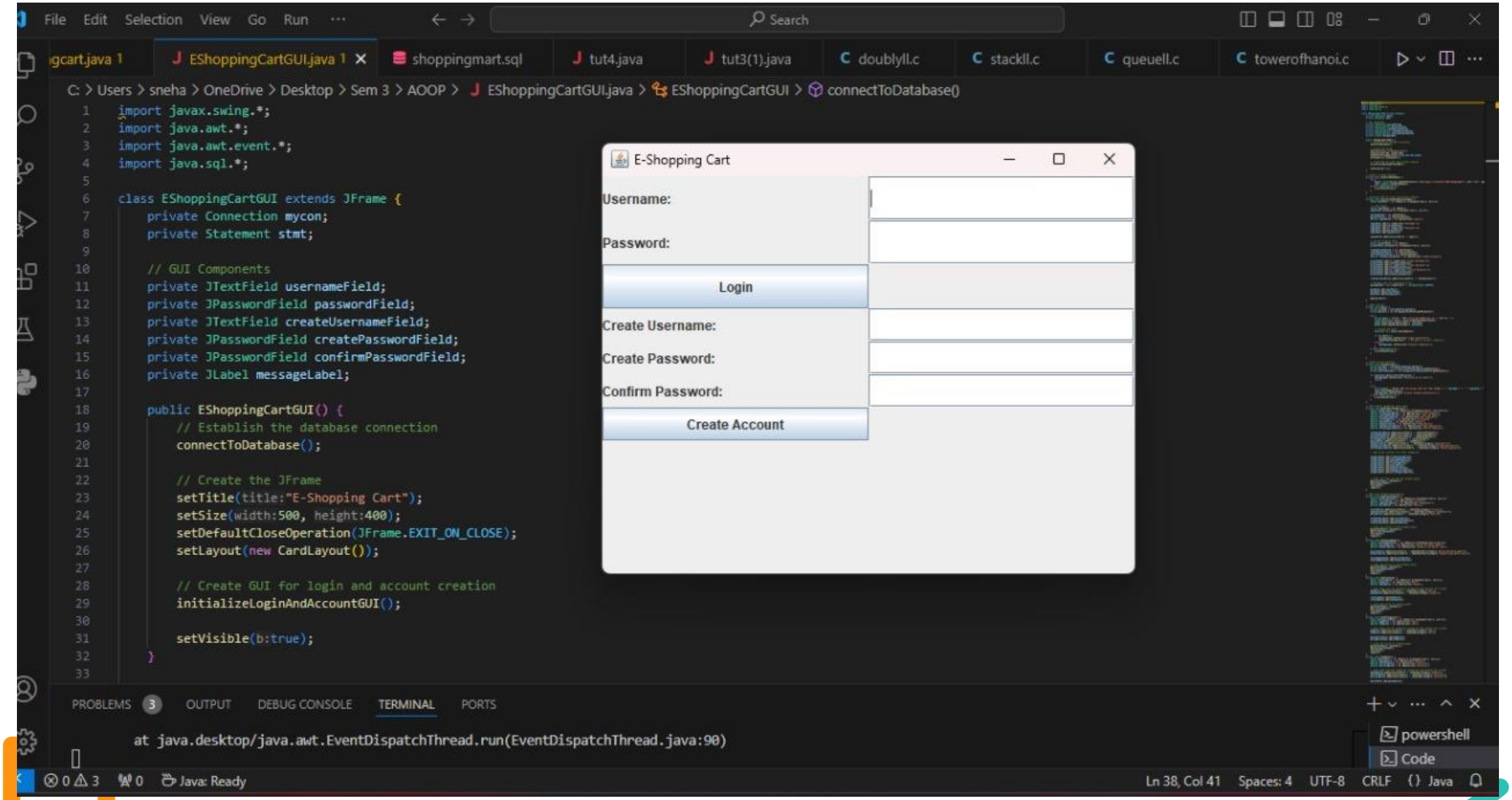
```java
// Add Items to Cart
private void addItemsToCart(String category) {
    try {
        String query = "SELECT pname FROM " + category;
        ResultSet rs = stmt.executeQuery(query);
        StringBuilder items = new StringBuilder();
        while (rs.next()) {
            items.append(rs.getString(columnLabel:"pname")).append(str:"\n");
        }

        String selectedItem = JOptionPane.showInputDialog(this, "Available items:\n" + items.toString() + "Enter item name to add to cart:");
        if (selectedItem != null && !selectedItem.trim().isEmpty()) {
            String cartQuery = "INSERT INTO add_to_cart SELECT * FROM " + category + " WHERE pname='" + selectedItem + "'";
            stmt.executeUpdate(cartQuery);
            JOptionPane.showMessageDialog(this, message:"Item successfully added to cart!");
        }
    } catch (SQLException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, message:"Error adding item to cart.");
    }
}
Run | Debug
public static void main(String[] args) {
    // Create an instance of the EShoppingCartGUI
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new EShoppingCartGUI();
        }
    });
}
}
```

```
19  CREATE TABLE fruits
20   ( pcode int,
21   pname varchar(20),
22   quantity varchar(20),
23   cost_in_Rs float);
24  INSERT INTO fruits VALUES(101, 'Apple', '1kg', 100);
25  INSERT INTO fruits VALUES(102, 'Orange', '1kg', 80);
26  INSERT INTO fruits VALUES(103, 'Pomegranate', '1kg', 120);
27  INSERT INTO fruits VALUES(104, 'Watermelon', '1', 60);
28  INSERT INTO fruits VALUES(105, 'Mango', '1kg', 100);
29  INSERT INTO fruits VALUES(106, 'Guava', '1kg', 110);
30  INSERT INTO fruits VALUES(107, 'Banana', 'dozen', 90);
31  INSERT INTO fruits VALUES(108, 'Dragonfruit', '1', 100);
32  INSERT INTO fruits VALUES(109, 'Muskmelon', '1', 80);
33  INSERT INTO fruits VALUES(110, 'Grapes', '1kg', 100);
34  SELECT * FROM fruits;
```

```sql
CREATE TABLE puzzle_and_boardgames
( pcode int,
 pname varchar(20),
 quantity varchar(20),
 cost_in_Rs float);
 INSERT INTO  puzzle_and_boardgames VALUES(201, 'Chess', '1', '599');
 INSERT INTO  puzzle_and_boardgames VALUES(202, 'Monopoly', '1', '2199');
 INSERT INTO  puzzle_and_boardgames VALUES(203, 'Jigsaw Puzzle', '1', '599');
 INSERT INTO  puzzle_and_boardgames VALUES(204, 'Ludo', '1', '299');
 INSERT INTO  puzzle_and_boardgames VALUES(205, 'Snakes and lader', '1', '399');
 INSERT INTO  puzzle_and_boardgames VALUES(206, 'Sudoku', '1', '499');
 INSERT INTO  puzzle_and_boardgames VALUES(207, 'Cards', '1', '159');
 INSERT INTO  puzzle_and_boardgames VALUES(208, 'UNO', '1', '119');
 INSERT INTO  puzzle_and_boardgames VALUES(209, 'Scrabble', '1', '799');
 INSERT INTO  puzzle_and_boardgames VALUES(210, 'Jenga', '1', '999');
 SELECT * FROM puzzle_and_boardgames;
```

```
CREATE TABLE men
( pcode int,
pname varchar(80),
quantity varchar(20),
cost_in_Rs float);
INSERT INTO  men VALUES(301,'Casual Tshirt-Black','1','1699');
INSERT INTO  men VALUES(302,'Casual Tshirt-White','1','1699');
INSERT INTO  men VALUES(303,'Formal Shirt-Beige','1','2899');
INSERT INTO  men VALUES(304,'Formal Shirt-SKy Blue','1','2899');
INSERT INTO  men VALUES(305,'Blue Straight Fit Denim Jeans','1','3499');
INSERT INTO  men VALUES(306,'Black Loose Fit Jeans','1','3799');
INSERT INTO  men VALUES(307,'Men Relaxed Fit Sweatpants-Grey','1','1199');
INSERT INTO  men VALUES(308,'Men Relaxed Fit Sweatpants-Beige','1','1199');
INSERT INTO  men VALUES(309,'Men SLim Fit Cargos-Grey','1','2399');
INSERT INTO  men VALUES(310,'Men SLim Fit Cargos-Black','1','2399');
SELECT * FROM men;
```


File Edit Selection View Go Run
Search
gcart.java 1
EShoppingCartGUI.java 1
shoppingmart.sql
tut4.java
tut3(1).java
doublyll.c
stackll.c
queuell.c
towerofhanoi.c
C: > Users > sneha > OneDrive > Desktop > Sem 3 > AOOP > EShoppingCartGUI.java > EShoppingCartGUI > connectToDatabase()
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
class EShoppingCartGUI extends JFrame {
private Connection mycon;
private Statement stmt;
// GUI Components
private JTextField usernameField;
private JPasswordField passwordField;
private JTextField createUsernameField;
private JPasswordField createPasswordField;
private JPasswordField confirmPasswordField;
private JLabel messageLabel;
public EShoppingCartGUI() {
// Establish the database connection
connectToDatabase();
// Create the JFrame
setTitle(title:"E-Shopping Cart");
setSize(width:500, height:400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new CardLayout());
// Create GUI for login and account creation
initializeLoginAndAccountGUI();
setVisible(b:true);
}
E-Shopping Cart
Username:
Password:
Login
Create Username:
Create Password:
Confirm Password:
Create Account
PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
at java.desktop/java.awt.EventDispatchThread.run(EventDispatchThread.java:90)
powershell
Code
0 3 0 Java: Ready
Ln 38, Col 41 Spaces: 4 UTF-8 CRLF Java

Conclusion and Future Enhancements

Hence in this mini project we try to we try to make an online shopping mart resolving the hassle of shopping offline. We have used different java libraries to make our project more interactive and user- friendly. Using tools simple computing tools like MySQL and java swing we have made a smaller scale version of online shopping businesses.

As for the future enhancement of this project, we would like to be able to add other features to our online shopping mart such as inventory history. In inventory history, we want to track the number of products that we have in our inventory before and after the user has ordered.

Moreover, we would like to make our shopping mart more secure by placing more security passes and checks throughout the program, which could be either by generating an OTP on the registered mobile number or by verifying the number. Implementing session management to track the logged-in user across multiple panels would add a level of security and convenience.

Adding functionality for checkout, payment, and order tracking would complete the e-commerce experience.

THANKS!!