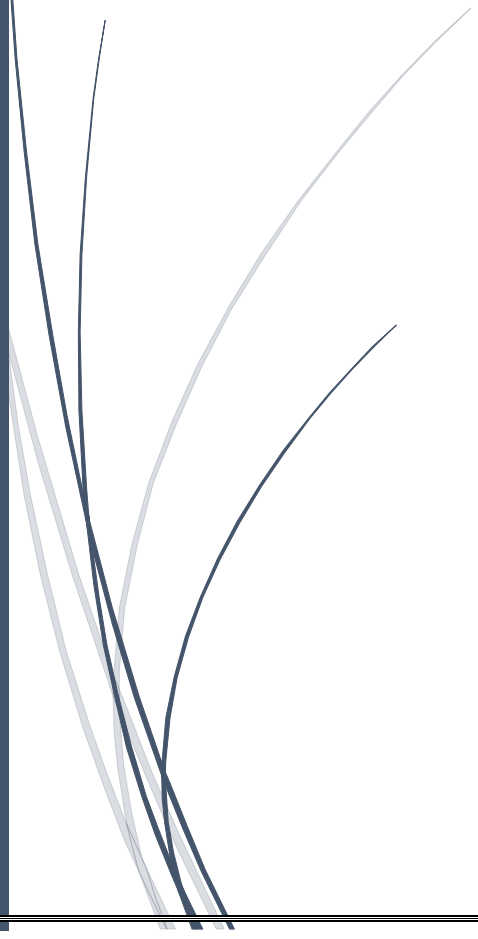
PROJECT REPORT

SPORTS SHOP SYSTEM

MANAGE YOUR SPORTS SHOP SMARTLY



By : Raj Agarwal

CONTENTS

1. **Introduction** ---------- 3

2. **Objective & Scope of the Project** ---------- 4

3. **Theoretical Background** ---------- 3

4. **Problem Definition & Analysis** ---------- 9

5. **System Implementation** ---------- 10

6.1 The Hardware used ---------- 10

6.2 The Softwares used ---------- 10

6. **System Design & Development** ---------- 11

7.2 Database Design ---------- 11

7.3 Menu Design ---------- 12

7.4 I/O Forms Design & Event Coding ---------- 12

7. **User Manual** ---------- 35

8.1 How to install ---------- 35

8.2 Working with Software ---------- 36

8. **References** ---------- 37

1. Introduction

This software project is developed to automate the functionalities of a sports shop. The purpose of the software project is to develop the Management Information System (MIS) to automate the record keeping of items, customers and orders with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a Sports Shop Billing System.

During coding and design of the software Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a powerful, open source RDBMS, MySQL is used as per requirement of the CBSE curriculum of Informatics Practices Course.

2. Objective & Scope of the Project

The objective of the software project is to develop a computerized MIS to automate the functions of a Sports Shop. This software project is also aimed to enhance the current record keeping system, which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

- ✓ To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.
- ✓ The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- ✓ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✓ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

In its current scope, the software enables user to retrieve and update the information from centralized database designed with MySQL . This software does not require much training time of the users due to limited functionality and simplicity.

During the development of Sports Shop Billing System project, Java NetBeans IDE, a powerful, open source event-driven form-based development environment is used for modular design and future expandability of the system.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software.

1. This software can store records and produce reports in pre-designed format in soft copy. There is no facility yet to produce customized reports. Only specified reports are covered.

So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

3. Theoretical Background

3.1 What is Database?

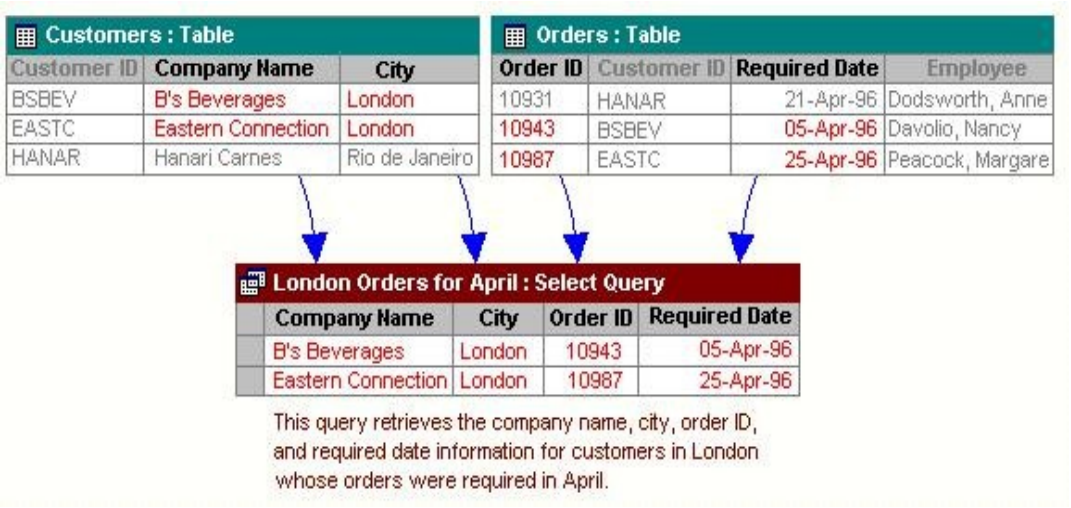
Introduction and Concepts:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc, you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables. You may and retrieve the data using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A Primary key is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A primary key is called foreign key when it is referred to from another table.

To find and retrieve just the data that meets conditions you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.



Role of RDBMS Application Program:

A computer database works as a electronic filing system, which has a large number of ways of cross-referencing, and this allows the user many different ways in which to re-organize and retrieve data. A database can handle business inventory, accounting and filing and use the information in its files to prepare summaries, estimates and other reports. The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it. Most of the database management systems have the following capabilities:

- Creating of a table, addition, deletion, modification of records.
- Retrieving data collectively or selectively.
- The data stored can be sorted or indexed at the user's discretion and direction.
- Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.
- Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.
- To maintain data integrity and database use.

The DBMS interprets and processes users' requests to retrieve information from a database. In most cases, a query request will have to penetrate several layers of software in the DBMS and operating system before the physical database can be accessed. The DBMS responds to a query by invoking the appropriate subprograms, each of which performs its special function to interpret the query, or to locate the desired data in the database and present it in the desired order.

3.2 What is MySQL ?

The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter, My. The name of the MySQL Dolphin (our logo) is “Sakila,”.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL is based on SQL.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License),

- **The MySQL Database Server is very fast, reliable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MySQL Server with other database managers on our benchmark page. MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

The Main Features of MySQL

- Written in C and C++.
- Works on many different platforms.
- Uses multi-layered server design with independent modules.
- Provides transactional and nontransactional storage engines.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Uses a very fast thread-based memory allocation system.
- Executes very fast joins using an optimized nested-loop join.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.
- Password security by encryption of all password traffic when you connect to a server.
- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections.
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.

3.3 What is NetBeans IDE ?

NetBeans started as a student project (originally called Xelfi) in the Czech Republic in 1996. The goal was to write a Delphi-like Java IDE in Java. Xelfi was the first Java IDE (Integrated Development Environment) written in Java, with its first pre-releases in 1997. Xelfi was a fun project to work on, especially since Java IDE space was uncharted territory at that time. The project attracted enough interest that these students, once they graduated, decided that they could market it as a commercial product. Soliciting resources from friends and relatives for a web space, they formed a company around it.

Soon after, they were contacted by Roman Stanek, an entrepreneur who had already been involved in several startups in the Czech Republic. He was looking for a good idea to invest in, and discovered Xelfi. He met with the founders; they hit it off, and a business was born.

In the spring of 1999, NetBeans DeveloperX2 was released, supporting Swing. The performance improvements that came in JDK 1.3, released in the fall of 1999, made NetBeans a viable choice for development tools. By the summer of 1999, the team was hard at work re-architecting DeveloperX2 into the more modular NetBeans that forms the basis of the software today.

Something else was afoot in the summer of 1999: Sun Microsystems wanted better Java development tools, and had become interested in NetBeans. It was a dream come true for the NetBeans team: NetBeans would become the flagship tool set of the maker of Java itself! By the Fall, with the next generation of NetBeans Developer in beta, a deal was struck. Sun Microsystems had also acquired another tools company, During the acqusition, the young developers who had been involved in open-source projects for most of their programming careers, mentioned the idea of open-sourcing NetBeans. Fast forward to less than six months later, the decision was made that NetBeans would be open sourced. While Sun had contributed considerable amounts of code to open source projects over the years, this was Sun's first *sponsored* open source project, one in which Sun would be paying for the site and handling the infrastructure.

Features of NetBeans

A free, open-source Integrated Development Environment for software developers. You get all the tools you need to create professional desktop, enterprise, web, and mobile applications with the Java platform, as well as C/C++, PHP, JavaScript, Groovy, and Ruby.

NetBeans IDE 6.9 introduces the JavaFX Composer, support for JavaFX SDK 1.3, OSGi interoperability, support for the PHP Zend framework and Ruby on Rails 3.0, and more.

4. Problem Definition & Analysis

The hardest part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirement. Defining and applying good, complete requirements are hard to work, and success in this endeavor has eluded many of us. Yet, we continue to make progress.

Problem definition describes the *What* of a system, not *How* . The quality of a software product is only as good as the process that creates it. Problem definition is one of the most crucial steps in this creation process. Without defining a problem, developers do not know what to build, customers do not know what to expect, and there is no way to validate that the built system satisfies the requirement.

Problem definition and Analysis is the activity that encompasses learning about the problem to be solved, understanding the needs of customer and users, trying to find out who the user really is, and understanding all the constraints on the solution. It includes all activities related to the following:

- ✓ Identification and documentation of customer's or user's needs.
- ✓ Creation of a document that describes the external behavior and the association constraints that will satisfies those needs.
- ✓ Analysis and validation of the requirements documents to ensure consistency, completeness, and feasibility
- ✓ Evolution of needs.

After the analysis of the functioning of a Sports Shop Billing System, the proposed System is expected to do the following: -

- ✓ To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for computerized Sports Shop Billing Shop.
- ✓ The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- ✓ To provide efficient and secured Information storage, flow and retrieval system, ensuring the integrity and validity of records.
- ✓ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✓ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

5. System Implementation

5.1 The Hardware used:

While developing the system, the used hardware are:

PC with Core i7 7th Gen processor 8GB RAM, SVGA and other required devices.

5.2 The Softwares used:

- Microsoft Windows® 10 as Operating System.
- Java NetBeans 6.9 as Front-end Development environment.
- MySQL as Back-end Sever with Database for Testing.
- MS-Word 365 for documentation.

6. System Design & Development

6.1 Database Design:

An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or contraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.
This software project maintains a database named **shopkeeper** which contains the following tables.

Table Design:

The database of Shopkeeper contains 3 tables. The tables are normalized to minimize the redundancies of data and enforcing the validation rules of the organization. Most of the tables are designed to store master records. The tables and their structure are given below.

Table: Shopkeeper

Column Name	*Type*	*Size*
Shopper _id (Primary Key)	Integer	100
name	Varchar	100
city	Varchar	100
phone	Varchar	11
address	Varchar	40

Table: Item

Column Name	*Type*	*Size*
Item_id (Primary Key)	Integer	100
Item_name	varchar	100
description	Varchar	1000
price	float	(11,2)

Table: Orderitem

Column Name	*Type*	*Size*
Orderno (Primary Key)	varchar	100
item_id	int	100
shopper_id	Varchar	100
quantity	int	100
Order date	Date	
price	float	(11,2)
discount	Float	(11,2)
amount	float	(11,2)

6.2 Menu Design:

JSS Infoware gateway comprises the following options, organized in a user friendly way. The menu system divided in Menu Bars, each having a pull down menus containing options for a specific task.

Sr.	Menu Bar	Pull Down Menu	Purpose	Forms Attached
1.	**Sports club**	**Customer entry**	Insertion of Customer records.	ShopINUI.java
		Item entry	Insertion of item records.	ItemINUI.java
		Order entry	Insertion of Order records.	OrderINUI.java
2.	**View/Edit**	**Customer list**	View Customer list	CustListUI.java
		Item list	View Item list	ItemListUI.java
		Order list	View Order list	OrderListUI.java
3.	**Quit**	**Application**	Close the Application.	--

6.3 I/O Forms Design & Event Coding:

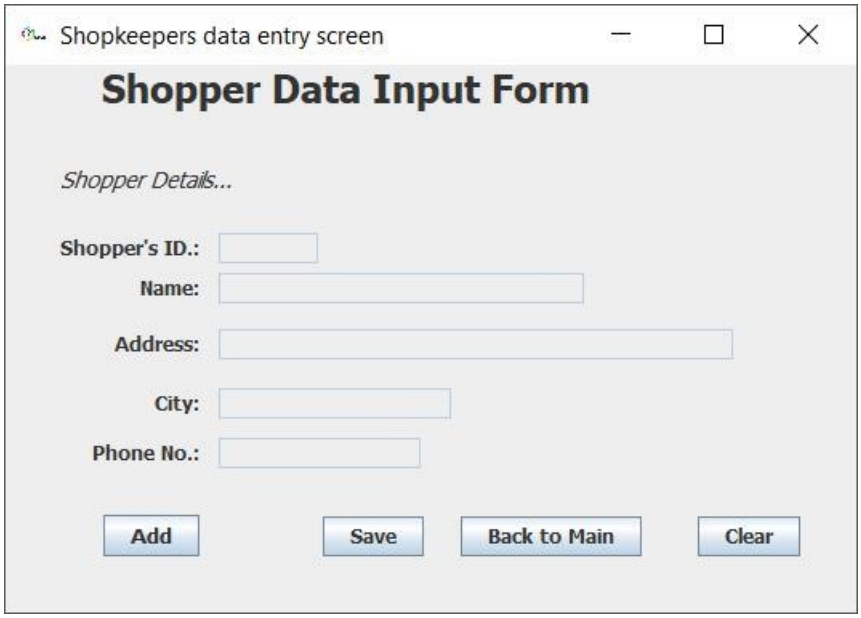
The software project for Sports Shop Billing System contains various forms along with programming codes. Forms (JFrames) and their event coding are given below.

Frame: MainMenuUI.java

Coding for MainMenuUI.java

```
import java.awt.Toolkit;

public class MainMenuUI extends javax.swing.JFrame {
   /** Creates new form MainMenuUI */
   public MainMenuUI() {
      initComponents();
      setIcon();
         }

 private void mnuCustAddActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new ShopINUI().setVisible(true);
}

   private void mnuItemAddActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new ItemINUI().setVisible(true);
}

   private void mnuOrderActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new OrderINUI().setVisible(true);
}

  private void ListCustActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new CustListUI().setVisible(true);
   }

   private void ListItemActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new ItemListUI().setVisible(true);

   }

   private void ListOrderActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new OrdListUI().setVisible(true);

   }

    private void QuitBtnActionPerformed(java.awt.event.ActionEvent evt) {
      System.exit(0);
       }
   private void setIcon() {
         setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
       }
```

Frame: ShopINUI.java



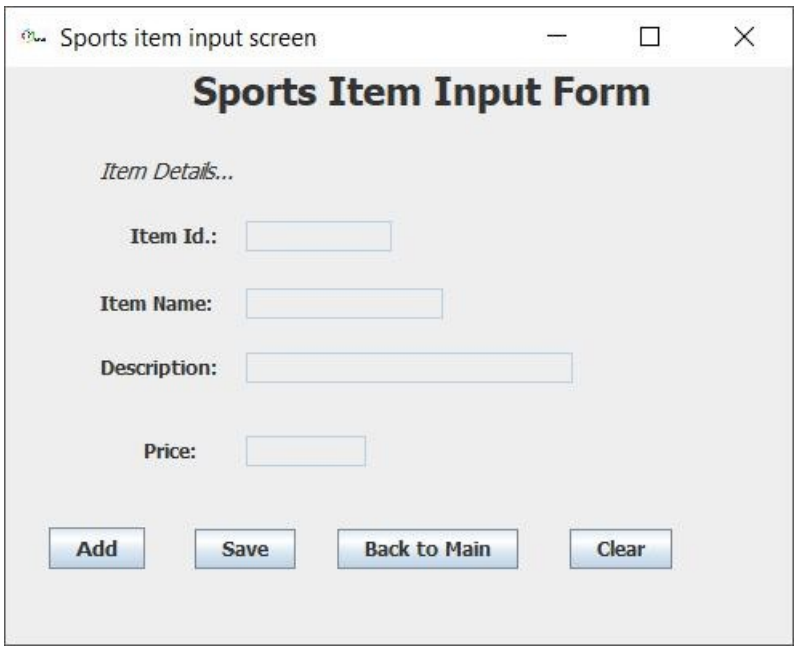
Coding of ShopINUI.java

```
import java.awt.Toolkit;
import java.sql.*;
import javax.swing.JOptionPane;

public class ShopINUI extends javax.swing.JFrame {

   /** Creates new form ShopINUI */
   public ShopINUI() {
      initComponents();
      setIcon();
   }

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
      this.setVisible(false);
      new MainMenuUI().setVisible(true);
}

   private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
      try {
         Class.forName("com.mysql.jdbc.Driver");
         Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
         Statement stmt = null;
         ResultSet rs = null;       // ResultSet for SHOPKEEPER table.
         String SQL = "SELECT * FROM shopkeeper";
```

```
            stmt = con.createStatement(); // Connection string for ResultSet - rs.
            rs = stmt.executeQuery(SQL);

            // Data transfer from JTextField control to variables
            String shno = txtSHno.getText().trim();
            String shname = txtSHName.getText();
            String shadd = txtSHAddress.getText();
            String shcity = txtSHCity.getText();
            double shphone = Double.parseDouble(txtSHPhone.getText());
            if (shno.length() <= 4) {
               String strSQL = "INSERT INTO shopkeeper(Shopper_id, Name, Address, City, Phone)
VALUES ('"+(shno)+"','"+(shname)+"','"+(shadd)+"','"+(shcity)+"','"+(shphone)+")";
               JOptionPane.showMessageDialog(this, "Record successfully inserted");
               int rowsEffected = stmt.executeUpdate(strSQL);
               System.out.println(rowsEffected + " rows effected");
            }
            else {
               JOptionPane.showMessageDialog(this, "Customer ID should not more than 4 character.");
            }
            con.close();
         } catch (Exception e) {
            JOptionPane.showMessageDialog(this,e.getMessage());

         }
}

   private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      txtSHno.setText("");
      txtSHName.setText("");
      txtSHAddress.setText("");
      txtSHCity.setText("");
      txtSHPhone.setText("");
}

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      try {
         Class.forName("com.mysql.jdbc.Driver");
         Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
         Statement stmt;
         ResultSet rs;        // ResultSet for shopkeeper table.
         String SQL = "SELECT * FROM shopkeeper";
         stmt = con.createStatement(); // Connection string for ResultSet - rs.
         rs = stmt.executeQuery(SQL);
         int shopid = 1;
```

```
        int sID=0;
        while (rs.next()) {
            sID = rs.getInt("shopper_id");
            shopid++;
        }
        sID++;
        shopid = sID;
        txtSHno.setText(Integer.toString(shopid));
        txtSHName.setEditable(true);
        txtSHAddress.setEditable(true);
        txtSHCity.setEditable(true);
        txtSHPhone.setEditable(true);

        con.close();
        rs.close();
        stmt.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}
private void setIcon() {
    setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
    }
```

Frame: ItemINUI.java

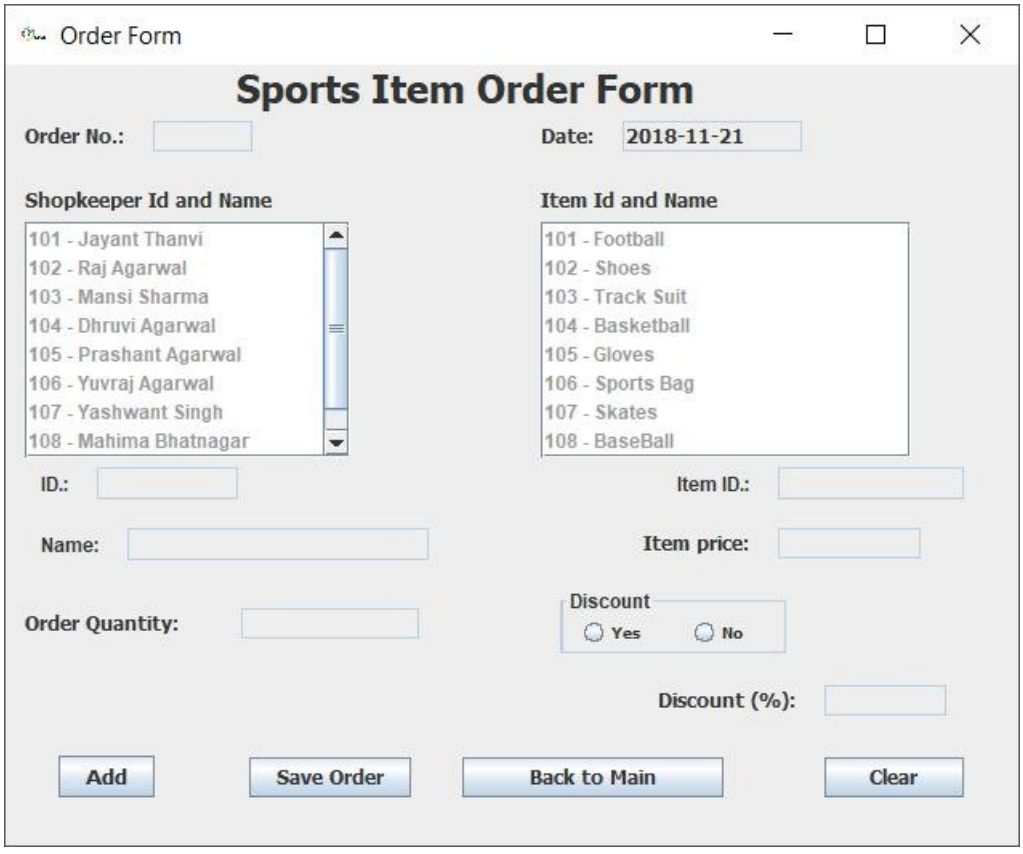


Coding for ItemINUI.java

```
import java.awt.Toolkit;
import java.sql.*;
import javax.swing.JOptionPane;

public class ItemINUI extends javax.swing.JFrame {

    /** Creates new form ItemINUI */
    public ItemINUI() {
        initComponents();
        setIcon();
    }
    private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
        //clearing the textfields
        txtItemno.setText("");
        txtItemName.setText("");
        txtItemDesc.setText("");
        txtItemPrice.setText("");
    }

    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            Statement stmt ;
            ResultSet rs ;        // ResultSet for item table.
            stmt = con.createStatement(); // Connection string for ResultSet - rs.

            // Data transfer from JTextField control to variables
            String itemno = txtItemno.getText();
            String itemname = txtItemName.getText();
            String desc = txtItemDesc.getText();
            double itemprice = Double.parseDouble(txtItemPrice.getText());
            String strSQL = "INSERT INTO item(Item_Id, Item_Name, Description, Price) VALUES
('"+(itemno)+"','"+(itemname)+"','"+(desc)+"','"+(itemprice)+"')";
            JOptionPane.showMessageDialog(this, "Record successfully inserted");
            stmt.executeUpdate(strSQL);
            con.close();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this,e.getMessage());
        }
}

    private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainMenuUI().setVisible(true);
}

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            Statement stmt;
```

```
        ResultSet rs;       // ResultSet for item table.
        String SQL = "SELECT * FROM item";
        stmt = con.createStatement(); // Connection string for ResultSet - rs.
        rs = stmt.executeQuery(SQL);
        int itemid = 1;
        int iID=0;
        while (rs.next()) {
           iID = rs.getInt("item_id");
           itemid++;
        }
        iID++;
        itemid = iID;
        txtItemno.setText(Integer.toString(itemid));
        txtItemDesc.setEditable(true);
        txtItemName.setEditable(true);
        txtItemPrice.setEditable(true);
        con.close();
        rs.close();
        stmt.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this,e.getMessage());

    }
}

private void setIcon() {
        setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
    }
```

Frame: OrderINUI.java



Coding for OrderINUI.Java

```
import java.awt.Toolkit;
import javax.swing.DefaultListModel;
import java.sql.*;
import java.text.SimpleDateFormat;
import javax.swing.JOptionPane;
import java.util.Date;

public class OrderINUI extends javax.swing.JFrame {

    /** Creates new form OrderINUI */
    public OrderINUI() {
        initComponents();
       jList1.setEnabled(false);
       jList2.setEnabled(false);
       showDate();
       setIcon();

    }
    void showDate(){
    Date d = new Date();
    SimpleDateFormat s = new SimpleDateFormat("yyyy-MM-dd");
    txtOrdDate.setText(s.format(d));
    }    private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {

        txtOrdno.setText("");
        txtOrdDate.setText("");
        txtSID.setText("");
        txtItemID.setText("");
        txtSName.setText("");
        txtItemPrice.setText("");
        txtOrdQty.setText("");
        txtDisc.setText("");
}

    private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainMenuUI().setVisible(true);
}

    private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            Statement stmt = null;
            ResultSet rs = null;        // ResultSet for order table.
            String SQL = "SELECT * FROM orderitem";
            stmt = con.createStatement(); // Connection string for ResultSet - rs.
            rs = stmt.executeQuery(SQL);

            float discP = 0; // Discount
            // Data transfer from JTextField control to variables
```

```
            String Ordno = txtOrdno.getText();
            String OrdDate = txtOrdDate.getText();
            String ItemID = txtItemID.getText();
            String ShopID = txtSID.getText();
            String ItemPrice = txtItemPrice.getText();
            double iPrice = Double.parseDouble(txtItemPrice.getText());
            String ordQty = txtOrdQty.getText();

            if (rdYes.isSelected()) {
               discP = Float.parseFloat(txtDisc.getText());
            }
            else {
               discP=0;
            }

            double amt = (Integer.parseInt(txtOrdQty.getText()) *
Double.parseDouble(txtItemPrice.getText())) - (Integer.parseInt(txtOrdQty.getText()) *
Double.parseDouble(txtItemPrice.getText()))*(discP*0.01);
            String strSQL = "INSERT INTO orderitem(orderno, OrderDate, Item_Id, Shopper_Id, Quantity,
price, discount, Amount ) VALUES
('"+(Ordno)+"','"+(OrdDate)+"','"+(ItemID)+"','"+(ShopID)+"',"+(ordQty)+","+iPrice+","+
discP+","+(amt)+")";
            JOptionPane.showMessageDialog(this, "Order successfully placed");
            int rowsEffected = stmt.executeUpdate(strSQL);
            System.out.println(rowsEffected + " rows effected");
            con.close();

         } catch (Exception e) {
            JOptionPane.showMessageDialog(this,e.getMessage());

         }
}

   private void formWindowGainedFocus(java.awt.event.WindowEvent evt) {
      // Creating a ListModel object sModel to perform DefaultListModel
      // method operations for Shopkeeper list
      DefaultListModel sModel = (DefaultListModel) jList1.getModel();

      // Creating a ListModel object iModel to perform DefaultComboBoxModel
      // method operations for Item list
      DefaultListModel iModel = (DefaultListModel) jList2.getModel();

      sModel.clear();
      iModel.clear();
      try {
         Class.forName("com.mysql.jdbc.Driver");
         Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
         Statement stmt;
         ResultSet rs;
         ResultSet rs1;

         String SQL = "SELECT * FROM shopkeeper";
         String SQL1 = "SELECT * FROM item";
         stmt = con.createStatement();
```

```
            // Steps to extract shopkeepers id and name
            rs = stmt.executeQuery(SQL);
            while (rs.next()) {
               String sID = rs.getString("Shopper_id");
               String Sname = rs.getString("Name");
               sModel.addElement(sID + " - " + Sname);
            }
            jList1.setModel(sModel);

            // Steps to extract item id and name
            rs1 = stmt.executeQuery(SQL1);
            while (rs1.next()) {
               String iID = rs1.getString("Item_Id");
               String Iname = rs1.getString("Item_Name");
               iModel.addElement(iID + " - " + Iname);
            }
            jList2.setModel(iModel);
            con.close();

         } catch (Exception e) {
            JOptionPane.showMessageDialog(this,e.getMessage());

         }

      }

      private void jList1MouseClicked(java.awt.event.MouseEvent evt) {
         // Extracting supplier id and name into a variable SidName
            String SidName = (String) jList1.getSelectedValue();
            String Sid = SidName.substring(0, 3);
            String Sname= SidName.substring(6);

            // Displays ID and name from ComboBox1
            txtSID.setText(Sid);
            txtSName.setText(Sname);
      }

      private void jList2MouseClicked(java.awt.event.MouseEvent evt) {
         try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            Statement stmt;
            ResultSet rs;

            // Extracting item id and item name into a variable ItemIDName
            String ItemIDName = (String) jList2.getSelectedValue();
            String ItemID = ItemIDName.substring(0, 3);
            String Iame= ItemIDName.substring(7);
            txtItemID.setText(ItemID);
            String SQL = "SELECT * FROM item where Item_Id = '"+(ItemID)+"'";
            stmt = con.createStatement();
            rs = stmt.executeQuery(SQL);

            while (rs.next()) {
               double iprice = rs.getDouble("Price");
```

```
            txtItemPrice.setText(Double.toString(iprice));
          }
          con.close();
        } catch (Exception e) {
          JOptionPane.showMessageDialog(this,e.getMessage());

        }

    }

    private void rdYesStateChanged(javax.swing.event.ChangeEvent evt) {
      if(rdYes.isSelected()==true){txtDisc.setEditable(true);}
    }

    private void rdNoStateChanged(javax.swing.event.ChangeEvent evt) {
       if(rdNo.isSelected()==true){txtDisc.setEditable(false);}
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
       try {
          Class.forName("com.mysql.jdbc.Driver");
          Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
          Statement stmt;
          ResultSet rs;        // ResultSet for order table.

          String SQL = "SELECT * FROM orderitem";

          stmt = con.createStatement(); // Connection string for ResultSet - rs.
          rs = stmt.executeQuery(SQL);

          int orderid = 1;
          int oID=0;
          while (rs.next()) {
             oID = rs.getInt("orderno");
             orderid++;
          }
          oID++;
          orderid = oID;
          txtOrdno.setText(Integer.toString(orderid));
          jList1.setEnabled(true);
          jList2.setEnabled(true);
          txtOrdQty.setEditable(true);

          con.close();
          rs.close();
          stmt.close();
        } catch (Exception e) {
          JOptionPane.showMessageDialog(this,e.getMessage());

        }}}

    private void setIcon() {
          setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
        }
```

Frame: CustListUI.java

Coding of CustListUI.java

```
import java.awt.Toolkit;
import javax.swing.table.*;
import java.sql.*;
import javax.swing.JOptionPane;

public class CustListUI extends javax.swing.JFrame {

   /** Creates new form CustListUI */
   public CustListUI() {
      initComponents();
      setIcon();
   }
 private void displayActionPerformed(java.awt.event.ActionEvent evt) {
      // Before writting the followng line, you should import the line:
      // import javax.swing.table.*; at the top of your application
      DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
      // Clear the existing table
      int rows = model.getRowCount();
      if (rows > 0) {
         for (int i = 0; i < rows; i++) {
            model.removeRow(0);
         }
      }
      // SQL Query
      String query = "SELECT * FROM shopkeeper";
      try {
         // Connect to MySQL database
         Class.forName("com.mysql.jdbc.Driver");
         Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
         // Create SQL statement and execute query.
         Statement stmt = con.createStatement();
```

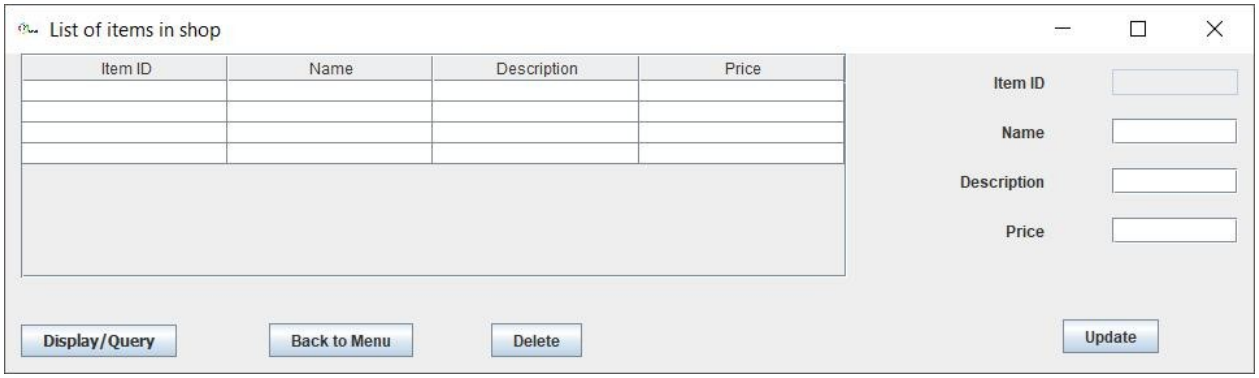
```
            ResultSet rs = stmt.executeQuery(query);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String Sid = rs.getString("Shopper_id");
                String SName = rs.getString("Name");
                String SAddress = rs.getString("Address");
                String SCity = rs.getString("City");
                String SPhone = rs.getString("Phone");
                model.addRow(new Object[] {Sid, SName, SAddress, SCity, SPhone});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
}

    private void backActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
          new MainMenuUI().setVisible(true);
}

    private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
        // Before writting the followng line, you should import the line:
        // import javax.swing.table.*; at the top of your application
        int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }
        // SQL Query
        String query = "delete from shopkeeper where shopper_id = "+cell+";";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            stmt.executeUpdate(query);
```

```
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
        // SQL Query
        String query1 = "SELECT * FROM shopkeeper";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query1);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String Sid = rs.getString("Shopper_id");
                String SName = rs.getString("Name");
                String SAddress = rs.getString("Address");
                String SCity = rs.getString("City");
                String SPhone = rs.getString("Phone");
                model.addRow(new Object[] {Sid, SName, SAddress, SCity, SPhone});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }

    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();

        // SQL Query
        String query = "select * from shopkeeper where shopper_id = "+cell+";";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            // Iterate through the result and display in textfield
            while (rs.next()) {
                cus_ID.setText(rs.getString("shopper_id"));
```

```
            name.setText(rs.getString("name"));
            city.setText(rs.getString("city"));
            ph.setText(rs.getString("phone"));
            add.setText(rs.getString("address"));
          }
        } catch (Exception e) {
          JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    private void updateActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
          for (int i = 0; i < rows; i++) {
            model.removeRow(0);
          }
        }

// SQL Query
        String query = "update shopkeeper set"
              +" address = '"+add.getText()
              +"' ,city = '"+city.getText()
              +"' ,phone = '"+ph.getText()
              +"' where shopper_id = "+cell+";";

        try {
          // Connect to MySQL database
          Class.forName("com.mysql.jdbc.Driver");
          Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
          // Create SQL statement and execute query.
          Statement stmt = con.createStatement();
          stmt.executeUpdate(query);



        } catch (Exception e) {
          JOptionPane.showMessageDialog(this, e.getMessage());
        }
        // SQL Query
        String query1 = "SELECT * FROM shopkeeper";
        try {
          // Connect to MySQL database
```

```
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
        // Create SQL statement and execute query.
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query1);

        // Iterate through the result and display on screen
        while (rs.next()) {
           String Sid = rs.getString("Shopper_id");
           String SName = rs.getString("Name");
           String SAddress = rs.getString("Address");
           String SCity = rs.getString("City");
           String SPhone = rs.getString("Phone");
           model.addRow(new Object[] {Sid, SName, SAddress, SCity, SPhone});
        }
      } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
      }}

   private void setIcon() {
        setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
      }
```

Frame: ItemListUI.java



Coding for ItemListUI.java

```
import java.awt.Toolkit;
import javax.swing.table.*;
import java.sql.*;
import javax.swing.JOptionPane;

public class ItemListUI extends javax.swing.JFrame {
```

```
    /** Creates new form ItemListUI */
    public ItemListUI() {
        initComponents();
        setIcon();
    }
private void backActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainMenuUI().setVisible(true);
}

    private void displayActionPerformed(java.awt.event.ActionEvent evt) {
        // Before writting the followng line, you should import the line:
        // import javax.swing.table.*; at the top of your application
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }
        // SQL Query
        String query = "SELECT * FROM Item";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String ITid = rs.getString("Item_id");
                String IName = rs.getString("Item_Name");
                String IDesc = rs.getString("Description");
                String IPrice = rs.getString("Price");
                model.addRow(new Object[] {ITid, IName, IDesc, IPrice});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
}

    private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
         // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }
        // SQL Query
        String query = "delete from Item where item_id = "+cell+";";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            stmt.executeUpdate(query);



        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
        // SQL Query
        String query1 = "SELECT * FROM Item";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query1);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String ITid = rs.getString("Item_id");
                String IName = rs.getString("Item_Name");
                String IDesc = rs.getString("Description");
                String IPrice = rs.getString("Price");
                model.addRow(new Object[] {ITid, IName, IDesc, IPrice});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
```

```
    }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();

        // SQL Query
        String query = "select * from item where item_id = "+cell+";";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            // Iterate through the result and display on screen
            while (rs.next()) {
                It_ID.setText(rs.getString("item_id"));
                name.setText(rs.getString("item_name"));
                desc.setText(rs.getString("description"));
                price.setText(rs.getString("price"));

            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    private void updtaeActionPerformed(java.awt.event.ActionEvent evt) {
        int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }

// SQL Query
        String query = "update item set"
                +" price = "+price.getText()
                +" ,item_name = '"+name.getText()
```

```
            +"' ,description = '"+desc.getText()
            +"' where item_id = "+cell+";";

        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            stmt.executeUpdate(query);



        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
        // SQL Query
        String query1 = "SELECT * FROM item";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query1);

            // Iterate through the result and display on screen
           while (rs.next()) {
                String ITid = rs.getString("Item_id");
                String IName = rs.getString("Item_Name");
                String IDesc = rs.getString("Description");
                String IPrice = rs.getString("Price");
                model.addRow(new Object[] {ITid, IName, IDesc, IPrice});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    private void setIcon() {
            setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
        }
```

Frame: OrderListUI.java

Coding for OrderListUI.java

```
import java.awt.Toolkit;
import javax.swing.table.*;
import java.sql.*;
import javax.swing.JOptionPane;


public class OrdListUI extends javax.swing.JFrame {

    /** Creates new form OrdListUI */
    public OrdListUI() {
        initComponents();
        setIcon();
    }

    private void backActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new MainMenuUI().setVisible(true);
}

    private void displayActionPerformed(java.awt.event.ActionEvent evt) {
        // Before writting the followng line, you should import the line:
        // import javax.swing.table.*; at the top of your application
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }
        // SQL Query
```

```
        String query = "SELECT * FROM orderitem";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String Ordno = rs.getString("Orderno");
                String Ordd = rs.getString("OrderDate");
                String OItemid = rs.getString("Item_Id");
                String OSid = rs.getString("Shopper_Id");
                String OQty = rs.getString("Quantity");
                String IPrice = rs.getString("price");
                String Disc = rs.getString("discount");
                String OAmount = rs.getString("Amount");
                model.addRow(new Object[] {Ordno, Ordd, OItemid, OSid, OQty, IPrice, Disc, OAmount});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
}

    private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
int row = jTable1.getSelectedRow();
        String cell = jTable1.getModel().getValueAt(row, 0).toString();
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        // Clear the existing table
        int rows = model.getRowCount();
        if (rows > 0) {
            for (int i = 0; i < rows; i++) {
                model.removeRow(0);
            }
        }
        // SQL Query
        String query = "delete from orderitem where orderno = "+cell+";";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver");
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
```

```
            stmt.executeUpdate(query);


        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }


        // SQL Query
        String query1 = "SELECT * FROM orderitem";
        try {
            // Connect to MySQL database
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/shopkeeper","root","jvst20032001");
            // Create SQL statement and execute query.
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery(query1);

            // Iterate through the result and display on screen
            while (rs.next()) {
                String Ordno = rs.getString("Orderno");
                String Ordd = rs.getString("OrderDate");
                String OItemid = rs.getString("Item_Id");
                String OSid = rs.getString("Shopper_Id");
                String OQty = rs.getString("Quantity");
                String IPrice = rs.getString("price");
                String Disc = rs.getString("discount");
                String OAmount = rs.getString("Amount");
                model.addRow(new Object[] {Ordno, Ordd, OItemid, OSid, OQty, IPrice, Disc, OAmount});
            }
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.getMessage());
        }
    }

    private void setIcon() {
        setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("icon.jpeg")));
        }
```

7. User Manual

7.1 How to Install Software:

Hardware Requirement-

- Intel Pentium/Celeron or similar processor based PC at Client/Server end.
- 128 MB RAM and 4GB HDD space (for Database) is desirable.
- Standard I/O devices like Keyboard and Mouse etc.
- Printer is needed for hard-copy reports.
- Local Area Network(LAN) is required for Client-Server Installation

Software Requirement-

- Windows 2000/XP OS is desirable.
- NetBeans Ver 5.1 or higher should be installed with JDK and JVM.
- MySQL Ver 6.1 with Library Database must be present at machine.

Database Installation

The software project is distributed with a backup copy of a Database named **shopkeeper** with required tables.The project is shipped with **SHOP.SQL** file which installs a database and tables in the computer system.

Note: The PC must have MySQL server with user (***root***) and password (***jvst20032001***) . If root password is any other password, it can be changed by running MySQL Server Instance Configure Wizard.

Start ▸ Program ▸ MySQL ▸ MySQL Server ▸ MySQL Server Instance Config Wizard

Provide current password of root and new password as “jvst20032001” , this will change the root password.

To install a MySQL database from a dump file ***(shop.sql)*** , simply follow the following steps.

Step 1: Copy the shop.sql file in **C:** folder.

Step 2: Open MySQL and type the following command to create the database named Library.

mysql> create database **shopkeeper**;

mysql> use **shopkeeper**;

mysql> source **C:/shop.sql**;

This will create a Library databse with required tables.

7.2 Working with SoftwareProject:

The Sports Shop Billing System consists of the following logically organised Menu-structure for the easy functionality. User may choose the menu options for corresponding works.

Sports Club:

This menu item gives options to insert record in Customer table, Item Table and order table.

View/Edit:

This menu gives options to delete, modify record in tables.

Quit:

This option shut down the program.

8. References

In order to work on this project titled ***-Sports Shop,*** the following books and literature are referred by me during the various phases of development of the project.

(1) The Complete Reference Java 2.0
-by Shildit

(2) MySQL, Black Book
-by Steven Holzner

(2) Understanding SQL
– Gruber

(3) http://www.mysql.org/

(4) http://www.netbeans.org/

(5) On-line Help of NetBeans ®

(6) Informatics Practices for class XII
-by Sumita Arora

(7) Together with Informatics Practices

(6) Various Websites of Discussion Forum and software development activities.

Other than the above-mentioned books, the suggestions and supervision of our teacher and our class experience also helped us to develop this software project.