CAMPUS MARKET PLACE

A MINI-PROJECT REPORT

Submitted by

KATHIR VIKAAS S **240701238**

HIRTHICK VASAN R **240701193**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **"Campus Market Place"** is the Bonafide work of **"KATHIR VIKAAS S, HIRTHIC VASAN R"** who carried out the project work under my supervision

SIGNATURE

Mrs.S.SATHIYAVATHI

ASSISTANT PROFESSOR SG

Dept. of Computer Science and Eng,

Rajalakshmi Engineering College

Chennai

This mini project report is submitted for the viva voce examination to be held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M. THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head of The Department **Dr. E.M. MALATHY** and our Deputy Head of the department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Mrs. S. SATHIYAVATHI** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of Computer science engineering.

1.KATHIR VIKAAS S

2.HIRTHICK VASAN R

ABSTRACT

The **Campus Marketplace** is a digital platform developed exclusively for college students to buy and sell pre-owned goods within their campus community. It provides an efficient and secure environment where students can exchange items such as books, gadgets, and accessories without the need for external applications.

The system allows users to **register, log in, list items for sale, browse available products, and purchase desired items** seamlessly. Built using **Java Swing** for the frontend and **MySQL** for the backend, the project ensures both usability and reliable data storage.

Through the **JDBC** connector, the application maintains smooth communication between the interface and the database. The project effectively demonstrates key principles of **Object-Oriented Programming (OOP)** and **Database Management Systems (DBMS)**. By encouraging reuse and student-to-student interaction, Campus Marketplace promotes sustainability and provides a convenient solution for managing second-hand goods within the campus environment.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Introduction

Campus Marketplace is a digital trading platform designed to help students buy and sell goods within their college campus. This project bridges the communication gap among students and provides a secure and efficient system for transactions.

1.2 Scope of the Work

The system provides an easy-to-use interface for students to register, login, and list items for sale. It also allows buyers to view all available items and purchase what they need. The project can be extended to multiple campuses and institutions in the future.

1.3 Problem Statement

Many college students struggle to sell or find second-hand items such as books, gadgets, and accessories. There is no centralized system that connects buyers and sellers within a campus. The Campus Marketplace solves this by providing a reliable internal marketplace.

1.4 Aim and Objectives of the Project

The main aim of the Campus Marketplace project is to create a simple, secure, and user-friendly system that allows students to exchange goods. Objectives include:

- Implementing user authentication using Java Swing and MySQL.
- Allowing users to list and browse available items.
- Facilitating item purchase and removal after sale.
- Maintaining data integrity using JDBC connections.

CHAPTER 2

SYSTEM REQUIREMENTS

Hardware Requirements:

• Processor: Intel i3 or higher

• RAM: 4GB or above

• Hard Disk: 500MB free space

Software Requirements:

• OS: Windows 10 or above

• IDE: IntelliJ IDEA / Eclipse

• Frontend: Java Swing

• Backend: MySQL

• Connector: MySQL JDBC Connector (.jar)

CHAPTER 3

MODULE DESCRIPTION

The application is structured into four primary modules: Data Model, Data Access Layer (DAO), Database Utility, and User Interfaces (UI).

3.1 DATA MODEL MODULE

• Item.java: Represents the blueprint for each item in the marketplace. It encapsulates fields like itemId, name, category, price, description, and sellerId. This class provides constructors, getters, and setters for managing data efficiently.

3.2 DATA ACCESS LAYER (DAO) MODULE

This module handles the database interaction logic. It includes:

• UserDAO.java – Manages user registration and login operations.

• ItemDAO.java – Handles CRUD operations for items, including insertion, retrieval, and deletion after purchase.

3.3 DATABASE UTILITY MODULE

• DBConnection.java: A utility class responsible for connecting the application to the MySQL database. It provides a static getConnection() method using JDBC. It ensures consistent access and efficient management of database connections.

3.4 USER INTERFACE (UI) MODULE

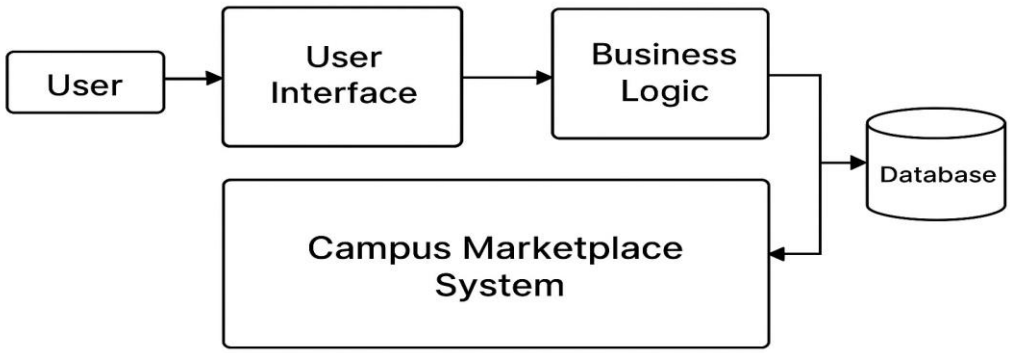
The UI module is built using Java Swing. It consists of interactive windows like Login, Registration, and Main Dashboard. • LoginUI.java – Handles authentication and user access.

• MainUI.java – Provides options to list, view, and buy items.

CHAPTER 4

SYSTEM DESIGN

The Campus Marketplace follows a client-server architecture where the Java Swing application (client) interacts with the MySQL database (server) through JDBC. The diagram below illustrates the overall structure of the system.



CHAPTER 5:

IMPLEMENTATION

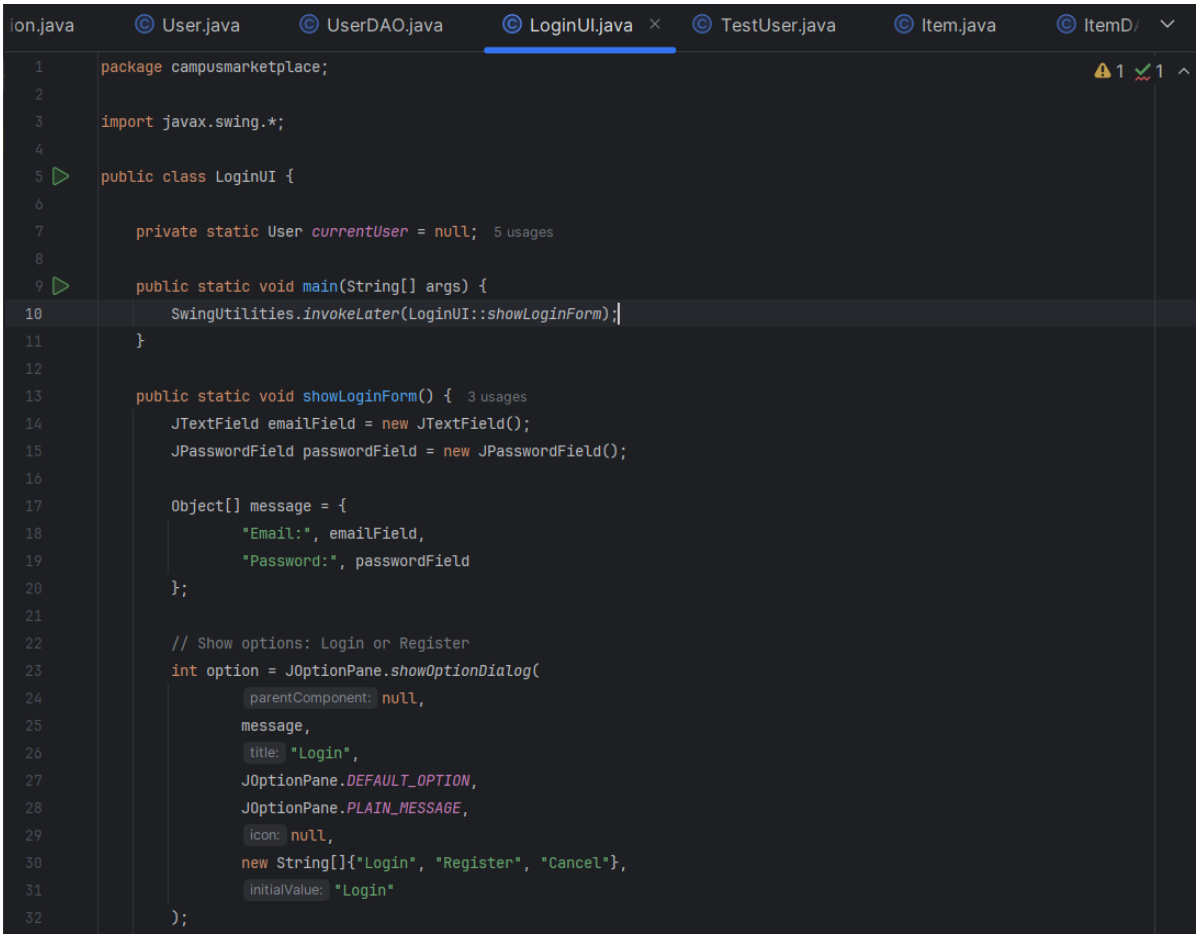
The implementation includes writing modular Java classes and establishing JDBC connections. DAO classes encapsulate the SQL queries. Swing components handle user interaction. Below sections are reserved for code snippets and class diagrams.

DBCONNECTION CODE:

DBConnection.java | User.java | UserDAO.java | LoginUI.java | TestUser.java | Item.java

```java
package campusmarketplace;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnection {

    // Make these static so static methods can access them
    private static final String url = "jdbc:mysql://localhost:3306/campus_marketplace?useSSL=false&serverTimezone=UTC";
    private static final String username = "root"; 1 usage
    private static final String password = "Password"; 1 usage

    static {
        try {
            Class.forName( className: "com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    // Static method to get DB connection
    public static Connection getConnection() throws SQLException { 6 usages
        return DriverManager.getConnection(url, username, password);
    }

    // Quick test
    public static void main(String[] args) {
        try (Connection conn = getConnection()) {
            if (conn != null && !conn.isClosed()) System.out.println("✅ DB Connected!");
            else System.out.println("❌ DB not connected.");
        } catch (SQLException e) {
            System.out.println("❌ Connection failed:");
            e.printStackTrace();
        }
    }
}
```

LOGINUI CODE:



USERDAO CODE:

DBConnection.java | User.java | UserDAO.java

```java
package campusmarketplace;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class UserDAO {  3 usages

    // Insert new user (with password)
    public static boolean insertUser(User user) {  2 usages
        String sql = "INSERT INTO Users (name, email, contact, password) VALUES (?, ?, ?, ?)";
        try (Connection conn = DBConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {

            pstmt.setString( parameterIndex: 1, user.getName());
            pstmt.setString( parameterIndex: 2, user.getEmail());
            pstmt.setString( parameterIndex: 3, user.getContact());
            pstmt.setString( parameterIndex: 4, user.getPassword()); // new password field

            int rows = pstmt.executeUpdate();
            return rows > 0; // true if insert was successful

        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }

    // Login method
    public static User login(String email, String password) {  1 usage
        String sql = "SELECT * FROM Users WHERE email = ? AND password = ?";
        try (Connection conn = DBConnection.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(sql)) {

            pstmt.setString( parameterIndex: 1, email);
            pstmt.setString( parameterIndex: 2, password);

            ResultSet rs = pstmt.executeQuery();
            if (rs.next()) {
                return new User(
                        rs.getInt( columnLabel: "userId"),
                        rs.getString( columnLabel: "name"),
                        rs.getString( columnLabel: "email"),
                        rs.getString( columnLabel: "contact"),
                        rs.getString( columnLabel: "password")
                );
            }

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null; // login failed
    }
}
```

CHAPTER 6

SCREENSHOTS

FIG 6.1 LOGIN PAGE

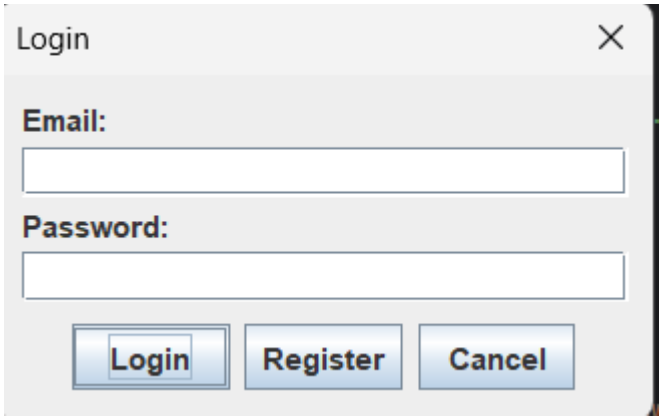


FIG 6.2 REGISTER PAGE

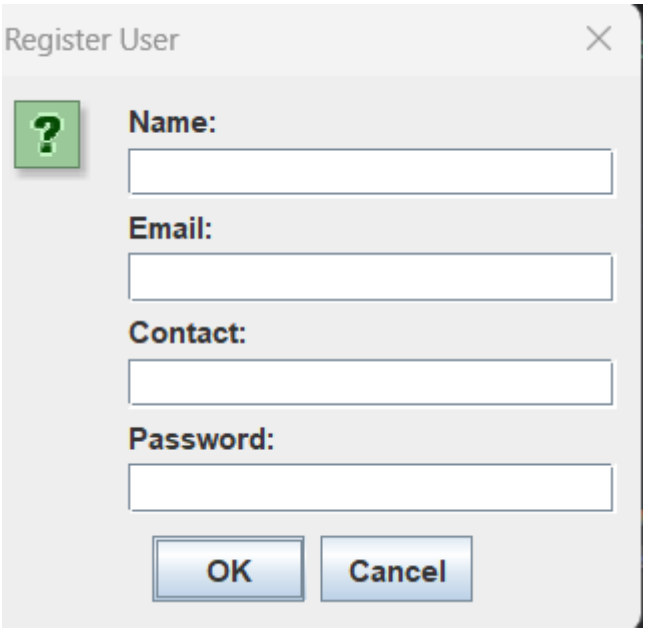


FIG 6.3 HOME PAGE

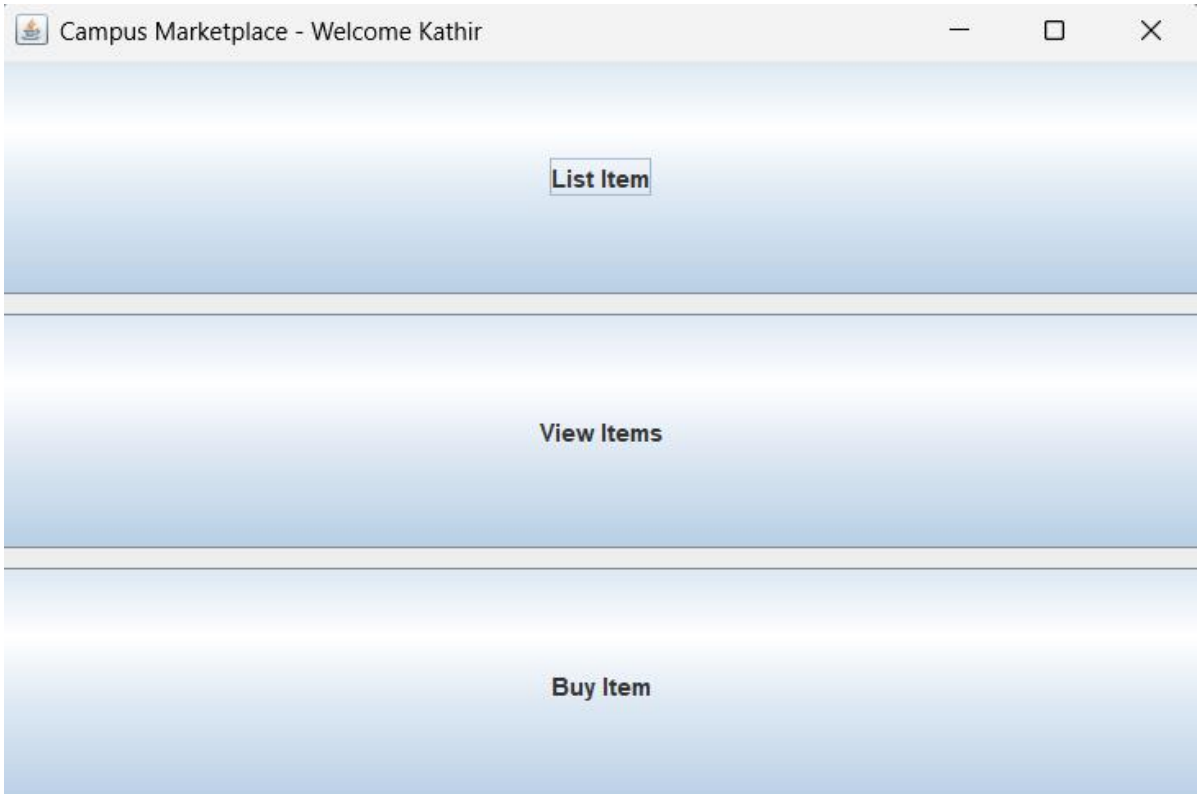


FIG 6.4 LIST ITEM PAGE

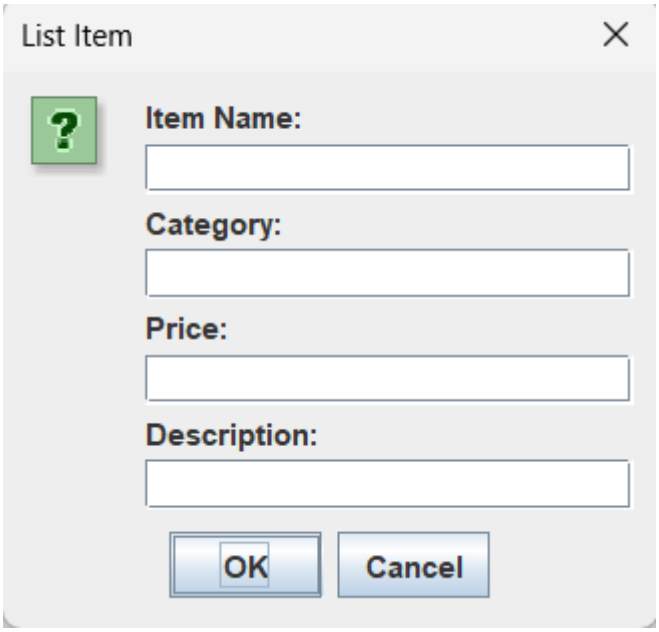


FIG 6.5 VIEW ITEMS PAGE

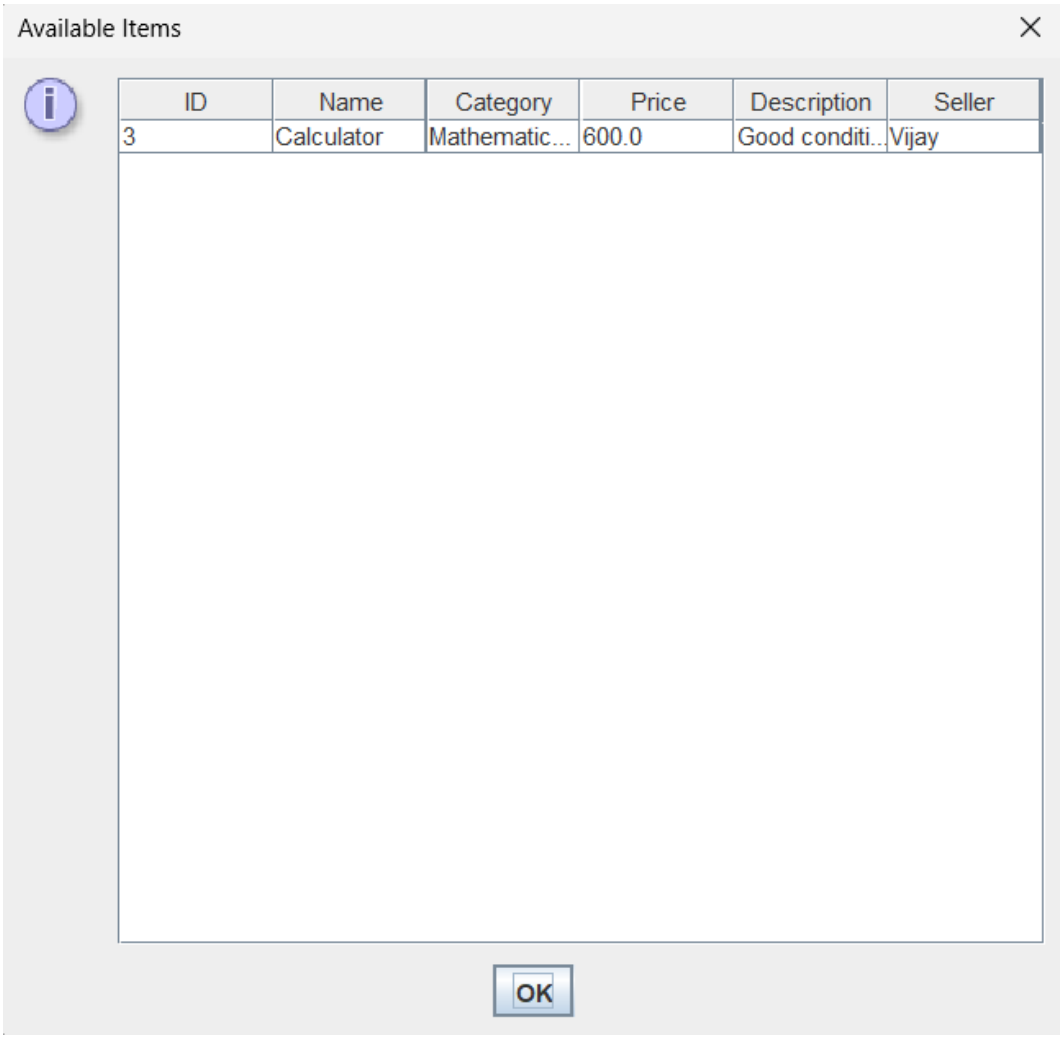
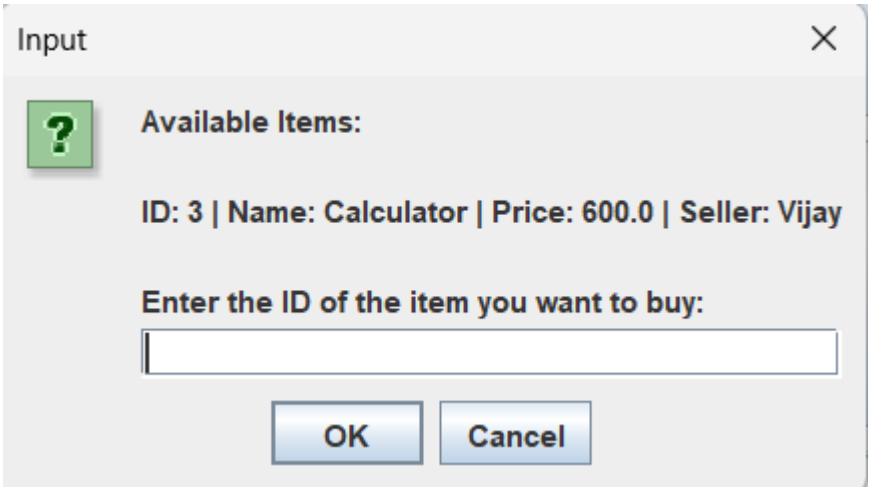


FIG 6.6 BUY ITEM PAGE



CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion:

The Campus Marketplace system provides an efficient and secure way for students to exchange goods. It demonstrates the practical application of Java and MySQL in creating an interactive software system.

Future Enhancements:

• Integrate chat between buyers and sellers.

• Add admin verification for item approval.

• Implement online payment gateway integration.

• Create a mobile application version for wider accessibility.

CHAPTER 8

REFERENCES

1. Java: The Complete Reference – Herbert Schildt
2. MySQL Documentation – https://dev.mysql.com/doc/
3. Oracle Java Tutorials – https://docs.oracle.com/javase/tutorial/
4. Online Java and JDBC resources.