**A**

**MINI PROJECT REPORT**

**ON**

**“ Musical Instrument System ”**

**SUBMITTED BY**

**Mr. Dhananjay Manik Bhagat**

**( SPPU Exam Seat No. 21714 )**

**D.Y. PATIL INSTITUTE OF MCA AND MANAGEMENT**

**AKURDI, PUNE-411044**

**Academic Year 2022-2023**

****Dr. D. Y. Patil Pratishthan’s

**D.Y. Patil Institute of Master of Computer Applications**

**And Management**

Sector No.29, Behind Akurdi Railway Station, Pradhikaran, Nigdi, Pune – 411044

Tel No: 020-27640998, 202737393, Fax no: 27653054, Website: www.dypimca.ac.in,

Email: director@dypimca.ac.in

**(Approved by AICTE, Recognized by DTE, Mah.; Affiliated to SPPU)**

**Ref:** DYPIMCAM/Acad/Note/ 35/23 Date: 16/03/2023

**CERTIFICATE**

This is to certify that the Project entitled

**“Musical Instrument System”**

Has been successfully completed

By

**Mr. Dhananjay Manik Bhagat**

Towards the partial fulfillment of

M.C.A. (Master of Computer Application)

Under

Savitribai Phule PuneUniversity for

Academic Year 2022-2023

**Ms. Vanita Patil**

**Internal Project Guide Internal Examiner**

**Dr. Kavita Suryawanshi Dr. K Nirmala**

**HOD-MCA, Vice Principal Director**

**External Examiner**

**Acknowledgement**

I would take the opportunity to thank **Dr. K. Nirmala, Director, DYPIMCA and Management** for her support, extended to me throughout the course.

I would like to thank **Dr. Kavita Suryawanshi, Vice Principal,** for her scholarly disposition, timely guidance, support and cooperation.

I would like to thank **Ms. Vanita Patil** for herkind guidance, keen interest, continuous encouragement and inspiration throughout the project work.

Finally, I gratefully thank all the faculty members of DYPIMCA and Management for their cooperation and support.

I also thankful to get constant encouragement, support and guidance from all Teaching and Non- Teaching Staff for their timely support which helped us in successfully completion of our project work.

Student Sign:

Student Name : Dhananjay Manik Bhagat

MCA I

Div- C

Roll No : 21714

**Dr. D. Y. Patil Pratishthan’s**

**D. Y. Patil Institute of Master of Computer Applications and Management**

**(M.C.A. Programme)**

(Approved by AICTE, New Delhi & Affiliated to Savitribai Phule Pune University)

**Dr. D. Y. Patil Educational Complex, Sector 29, Pradhikaran, Akurdi, Pune – 411 044**Tel No: (020)27640998, Website: www.dypimca.ac.in, E-mail : [director@dypimca.ac](mailto:director@dypimca.ac).

**Index**

|  |  |  |
| --- | --- | --- |
| Sr. No | Chapter Name | Page No |
| **1** | **Chapter 1: Introduction** | 5 |
| 1.1 | Project Objectives | 6 |
| 1.2 | Existing System and Need of System | 7 |
| 1.3 | Scope of work | 8 |
| 1.4 | Operating Environment-Hardware and Software | 9 |
| 1.5 | Technology Used | 9 |
| 1.6 | Module Specification | 10 |
| **2** | **Chapter 2: Analysis & Design** | 11 |
| 2.1 | Data Flow Diagram | 11 |
| 2.2 | Entity Relationship Diagram | 12 |
| 2.3 | Use Case Diagram | 13 |
| 2.4 | Activity Diagram | 14 |
| 2.5 | Sequence Diagram | 15 |
| 2.6 | Class Diagram | 16 |
| 2.7 | User Interface Screens | 17-23 |
| 2.8 | Table Structure | 24-26 |
| **3** | **Drawbacks and Limitations** | 27 |
| **4** | **Proposed Enhancement** | 28 |
| **5** | **Conclusion** | 29 |
| **6** | **Bibliography** | 30 |

1. **Introduction**

* The project “MUSICAL INSTRUMENTS SYSTEM” is an offline windows-based application designed to manage all purchase, sales related operations within an organization.
* The main objective of the application is to automate the existing system of manually maintaining the records of the product, customer, vendor, shop, purchase, sales details etc. In this project once the information is inputted, it will provide quick information regarding data.
* The new system will increase data accuracy, make musical instruments system more secure, effective, convenient and accessible and begin to coordinate information across the system.
* It gives a brief look of the products being purchased from the vendor and the products being sold to the customer and reports are generated accordingly.
* The main scope of our Musical Instrument System is that, it’s a user-friendly application and this project is used to give maximum information to the user about product purchased from vendor and product sold to the customer, purchase and sales details etc.
* This project is very convenient for the user as it minimizes the process work as well as the tedious job.
* Data approach and access made easier and convenient as it provides easy and quick access to particular product and services. Also, time required for accessing any detail will be very less. Hence the system saves time, efforts and cost.
  1. **Project Objective**
* Provide the good user interface to ease of use and it also provide the security to the database.
* The System can view the details of any record.
* To allow only authorized user to access various function and processed available in the System.
* Locate any instruments information wanted by the user.
* Reduced clerical work as most of the work done by computer.
* Provide greater speed and reduced time consumption.

**1.2 Existing System and Need of System**

* Present System is totally depending on manual data management for different activities such as maintaining details of the product, customer, vendor etc.
* They maintain different register for various activities such as storing and searching data which is tedious and time-consuming task.
* In present system all transactions are done manually with pen and paper. So, the frequent updating is not possible also generating reports, accurately is not possible with current system.
* Existing system is lacking in facilities such as searching, deleting, and updating data efficiently and effectively.

**1.3 Scope of Work**

* Manage online Shopping Easily.
* Secure Registration and Profile management facilities for customer.
* Easy and quick access to particular product and services.
* This system is easy in handle and user friendly.
* Time required for accessing any detail will be very less.
* User can view details of the parts without going anywhere.
* It is convenient for users as this system provides accurate costs and description of the system.
* User can view different categories of product of different brands at a single place.
* This system calculates bill instantly and user can pay online.
* The system calculate bill instantly and user can pay online.

Hence the system saves time, efforts and cost.

**1.4 Operating Environment – Hardware and Software**

* **Hardware Requirements :**
* Processor: Intel i3 10th Generation & above.
* Hard Disk: 25 GB Minimum.
* RAM: 1 GB Minimum.

* **Software Requirements :**
* Operating System: Windows 7 and Higher Version.
* Front – End: Java Servlet/JSP, Bootstrap, HTML, JavaScript.
* Back - End: Apache Tomcat
* Database: MySQL.
* Editor: Eclipse IDE

**1.5 Technology Used**

* With the advent of latest technology if we do not update our system then our business result in losses gradually with time.
* Here, Java Technology – Servlet/JSP is used for logic and view purpose. For designing HTML/CSS/Bootstrap/JavaScript/jQuery is used.
* For Database MySQL 8.0 version is used.

**1.6 Module Specification**

* **ADMIN LOGIN MODULE**:

This module is used to login to the software. The admin has to enter the username and password in order to login to the application.

* **PRODUCT MODULE**:

This module used to add product details related to purchase and sales.

* **CUSTOMER MODULE**:

This module contains the details of customer who has purchased the products and details related to it in a systematic way.

* **VENDOR MODULE**:

This module contains the details of all the vendors from whom the product has been purchased by the shopkeeper.

* **SHOP MODULE**:

This module contains details of the shop.

* **PURCHASE DETAILS**:

It deals with purchase like, shopkeeper purchasing the products from vendor.

* **SALES DETAILS**:

It deals with sales like, shopkeeper selling the products to customer.

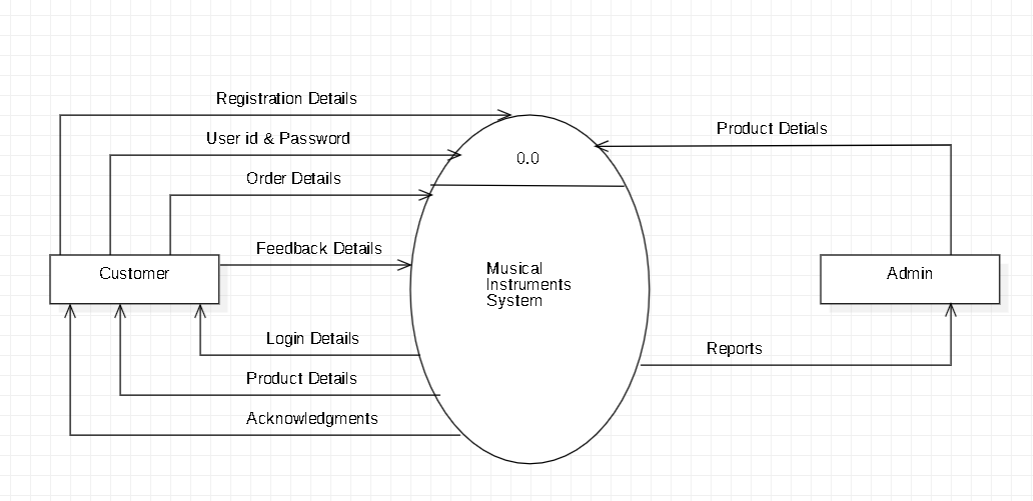
* **REPORT GENERATION**:

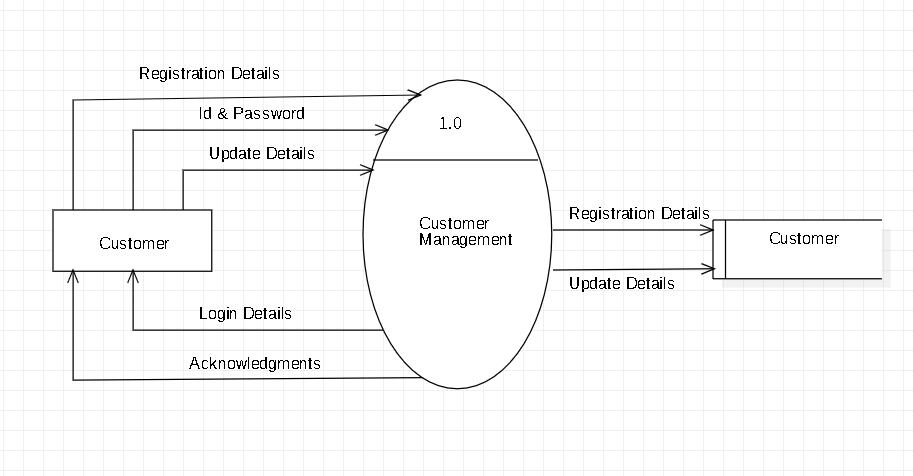
This module provides a way for viewing the data of sales.

**2.Analysis & Design**

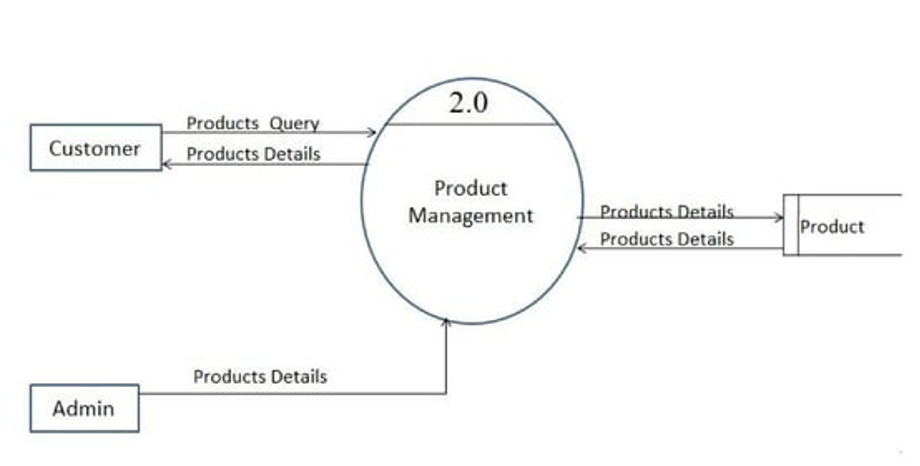
* 1. **Data Flow Diagram**

**Level 0 :**

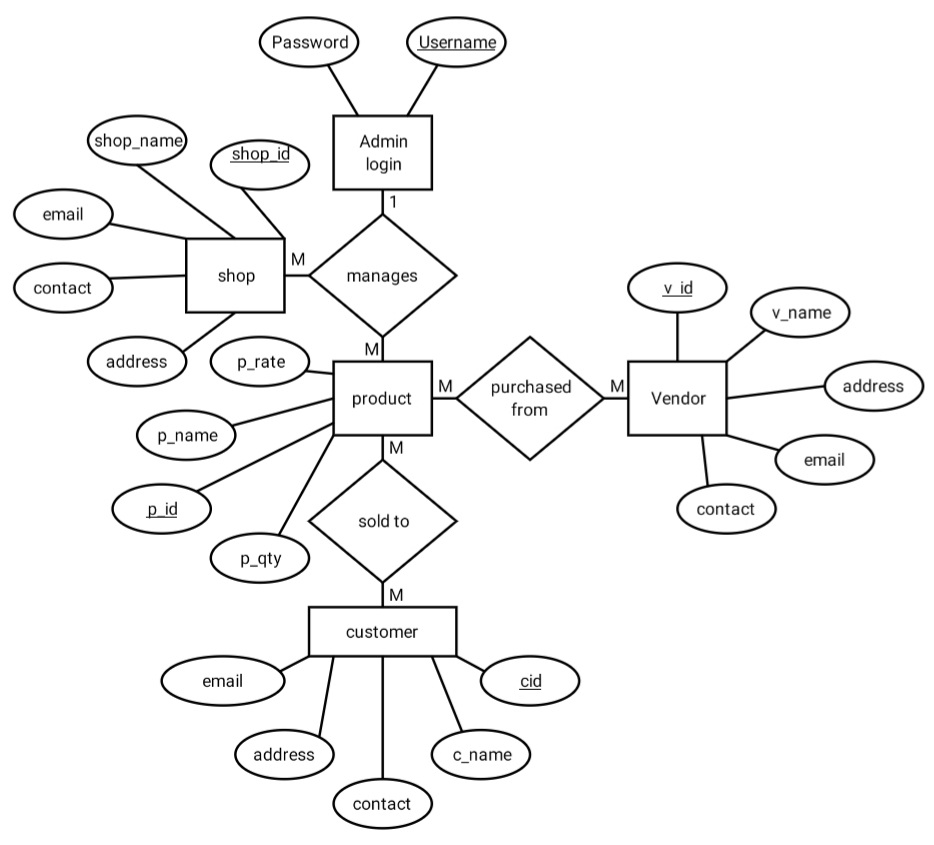
****

**Level 1 : **

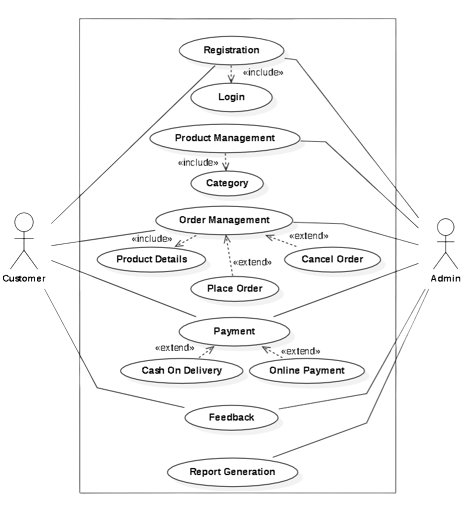
**Level 2 :**

****

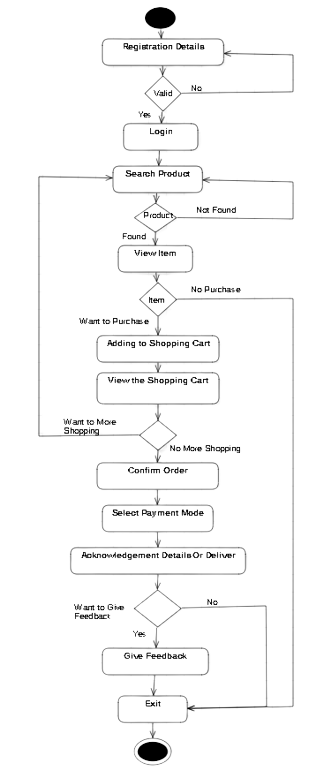
* 1. **Entity Relationship Diagram**

****

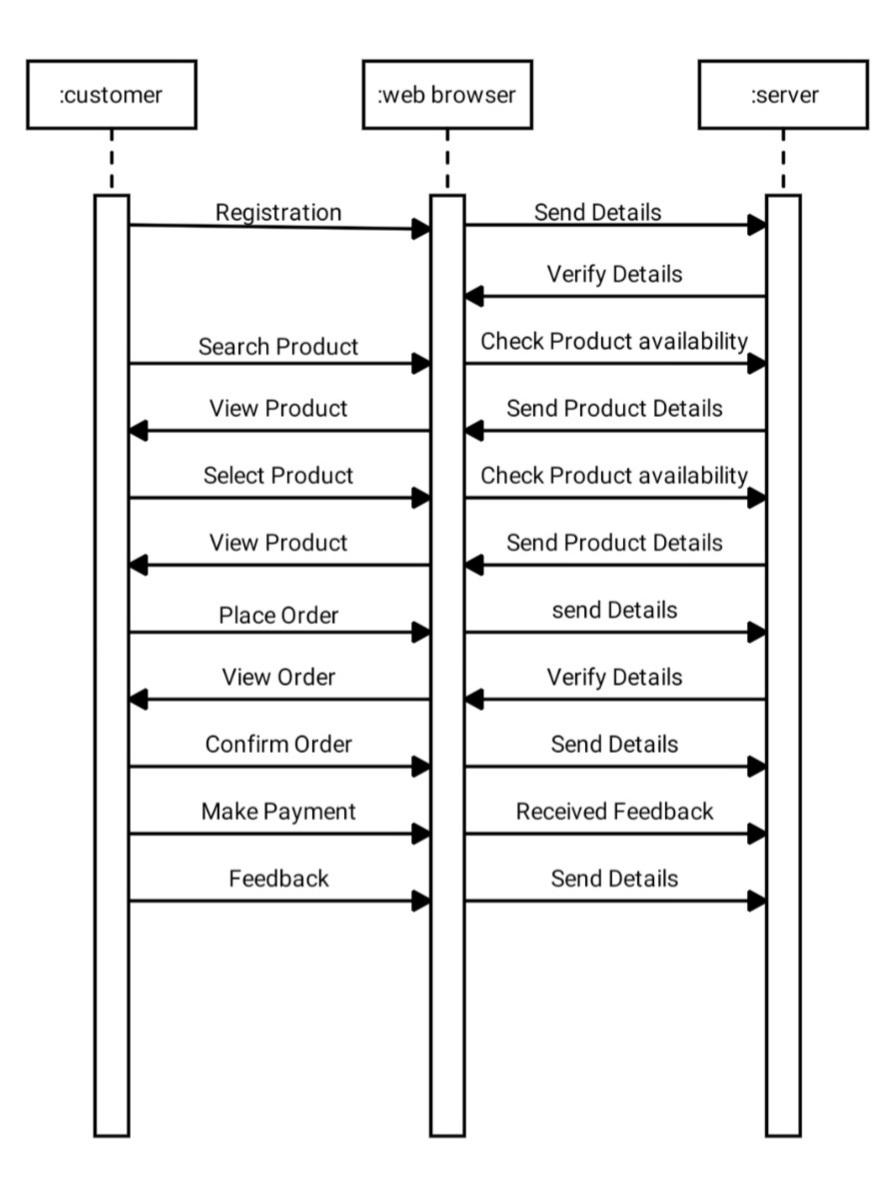
**2.3 Use Case Diagram**

****

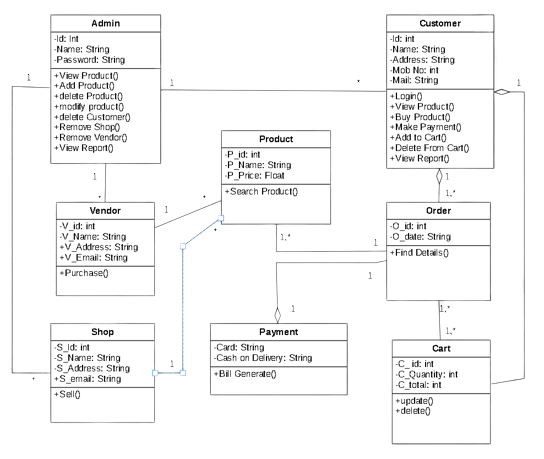
**2.4 Activity Diagram**

****

**2.5 Sequence Diagram**

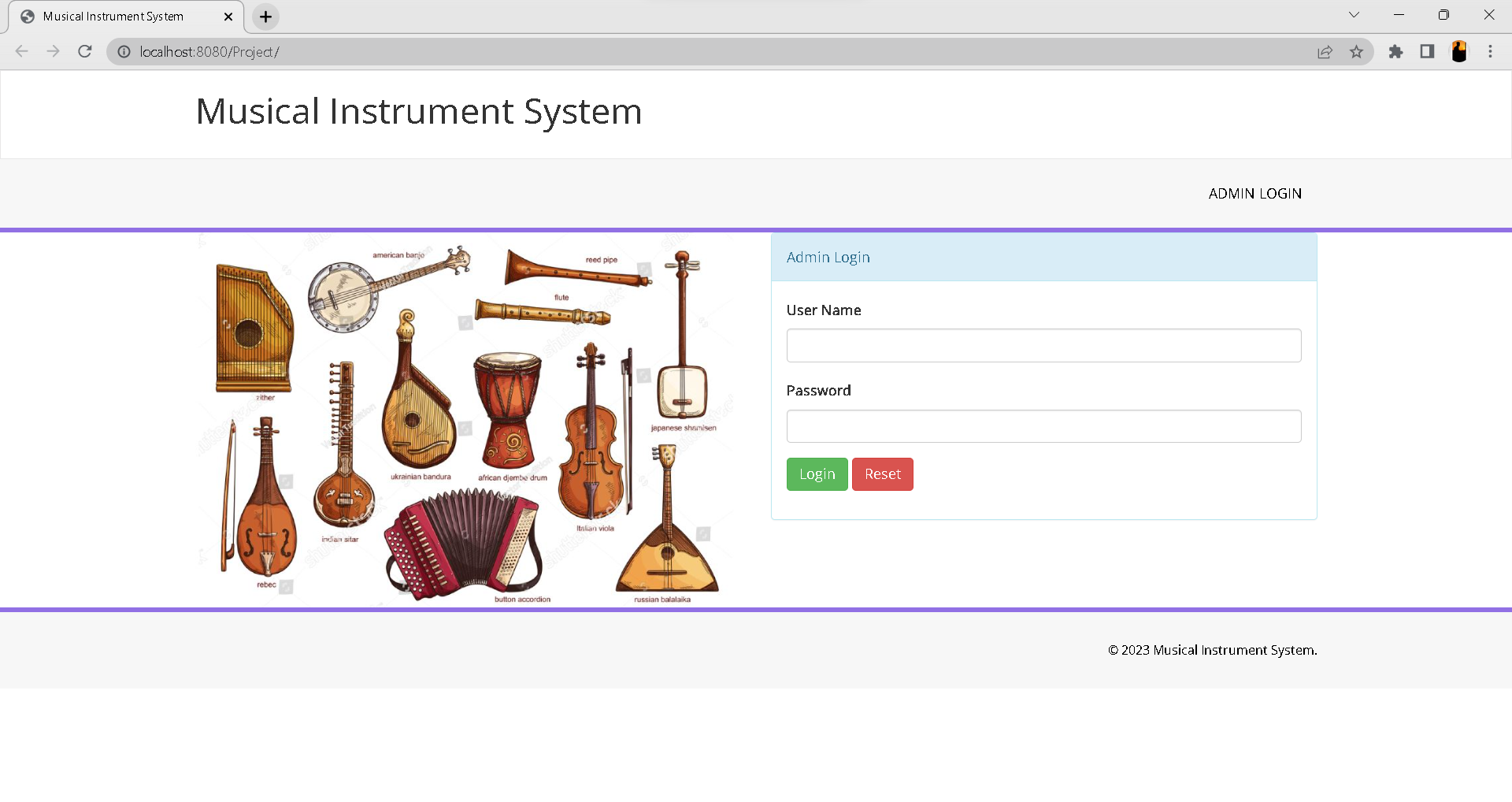
****

**2.6 Class Diagram**

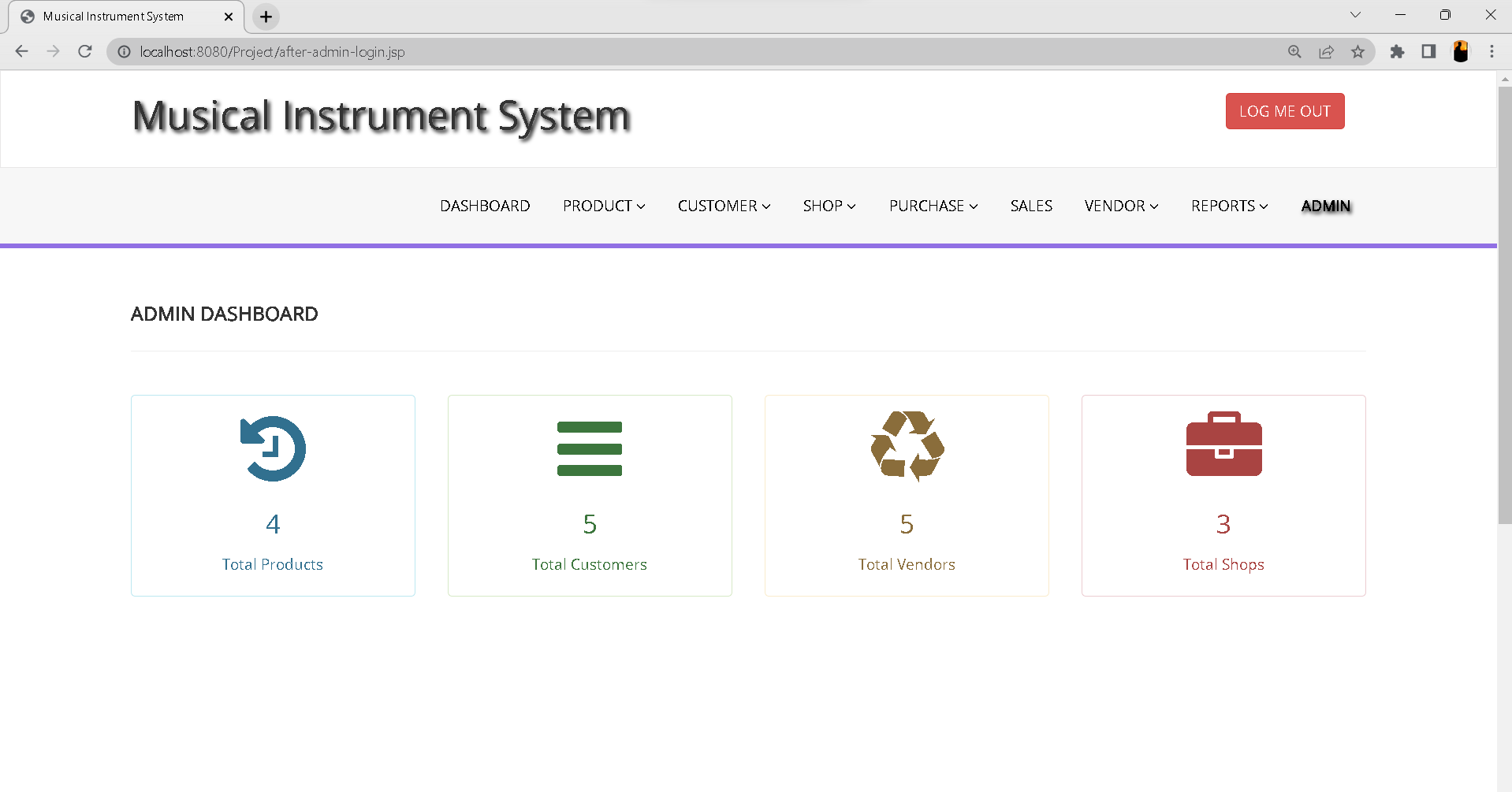
****

**2.7 User Interface Screens**

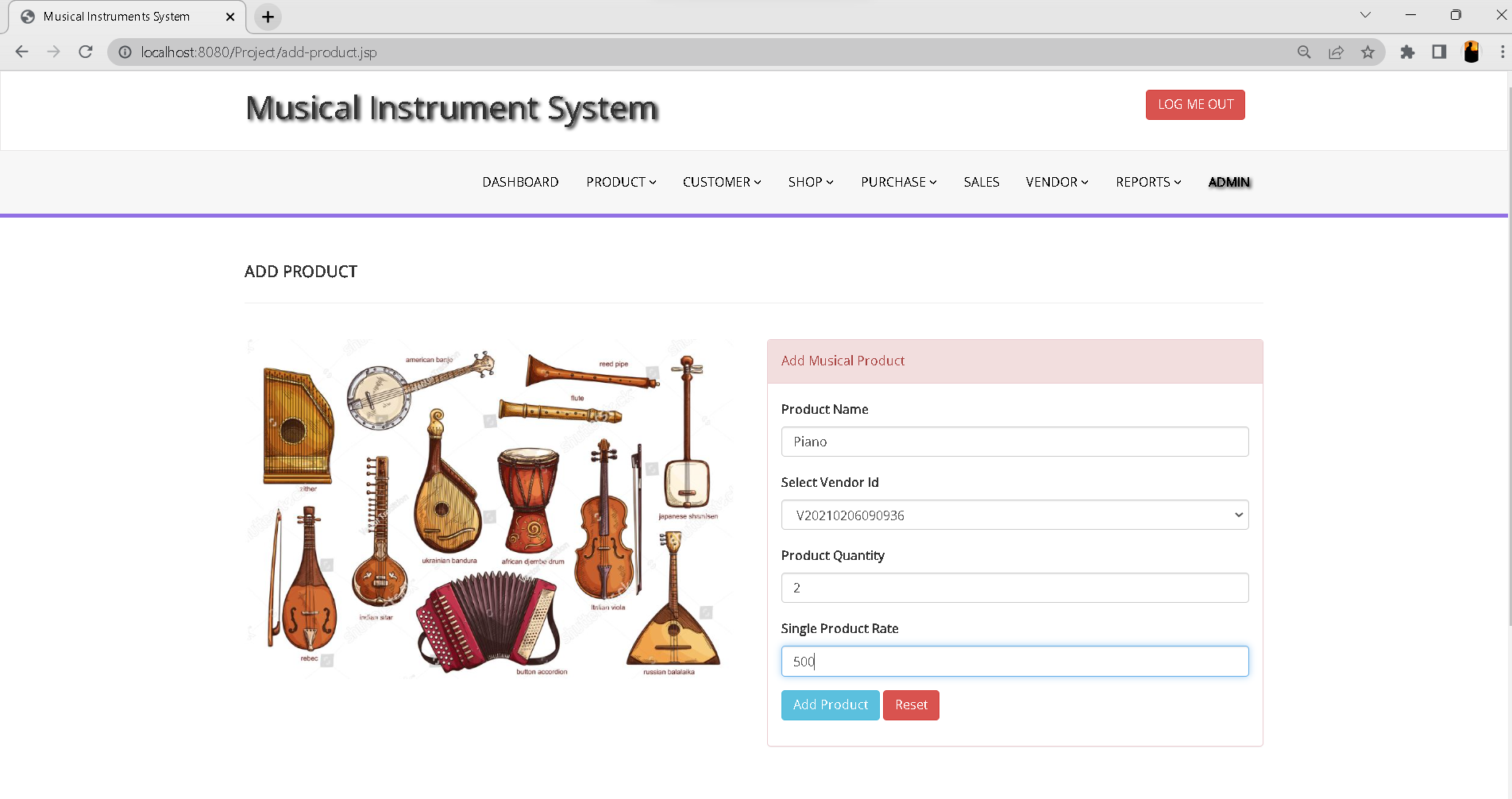
* **Admin Login :**

****

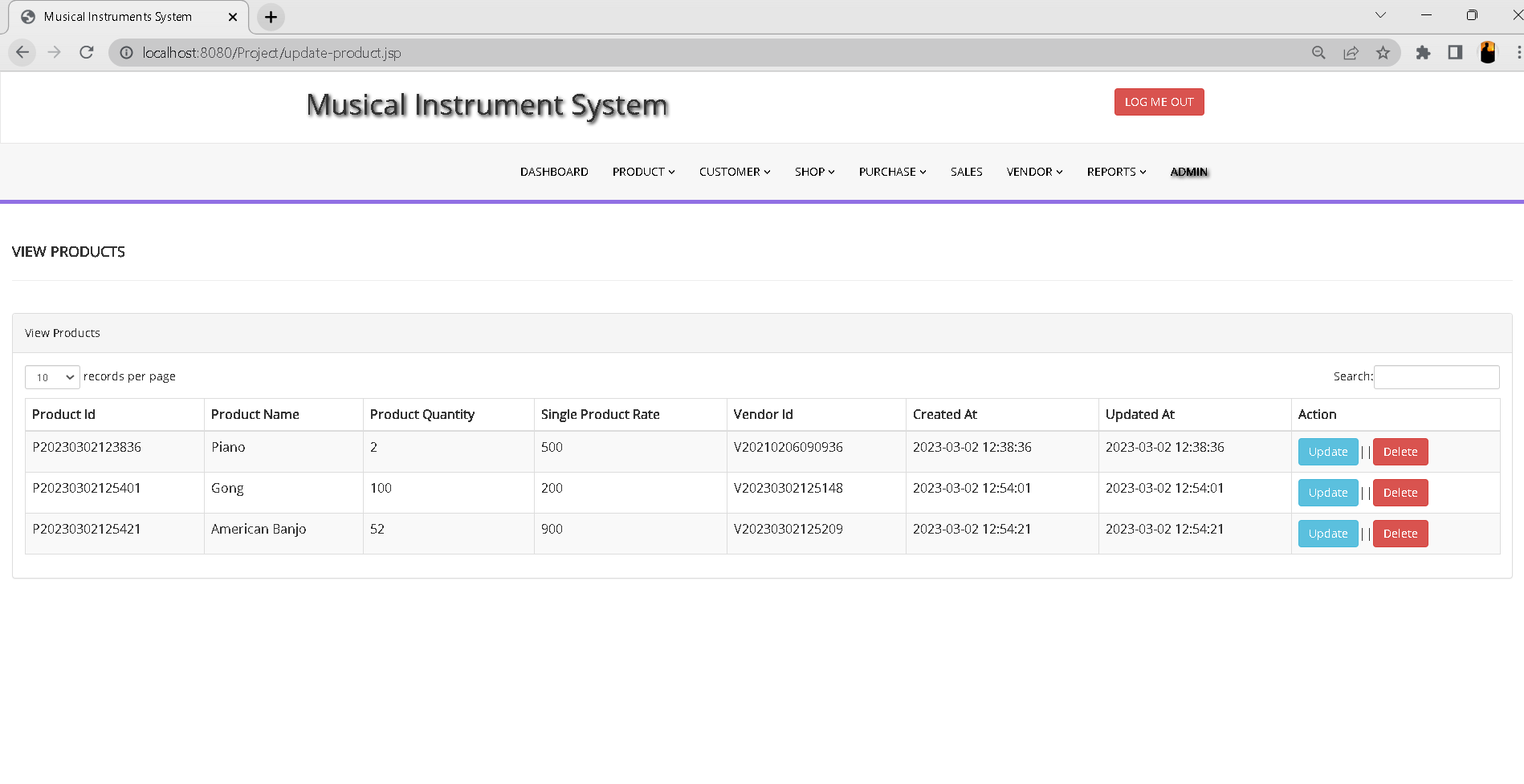
* **Dashboard :**

****

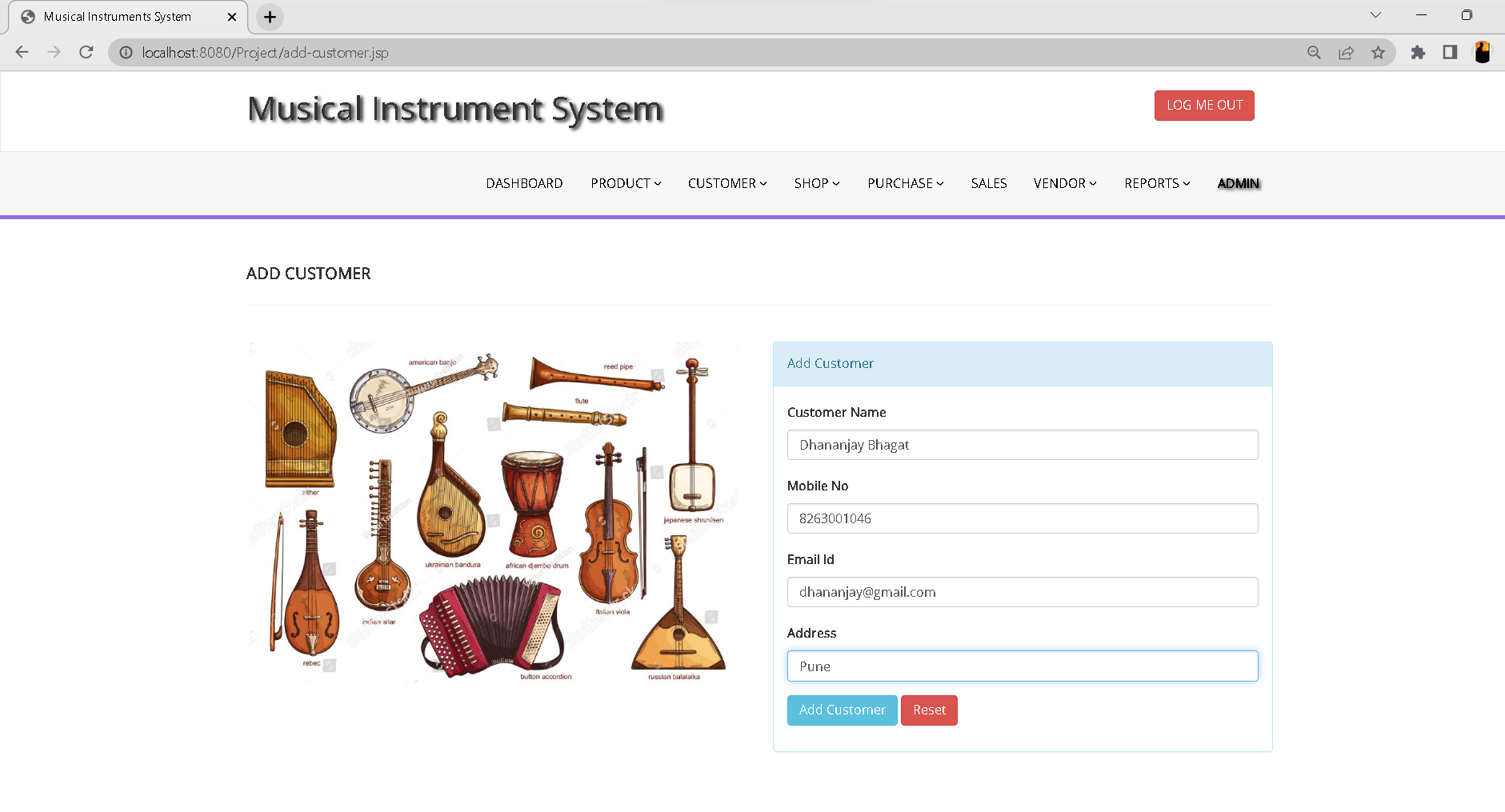
* **Add Product :**

****

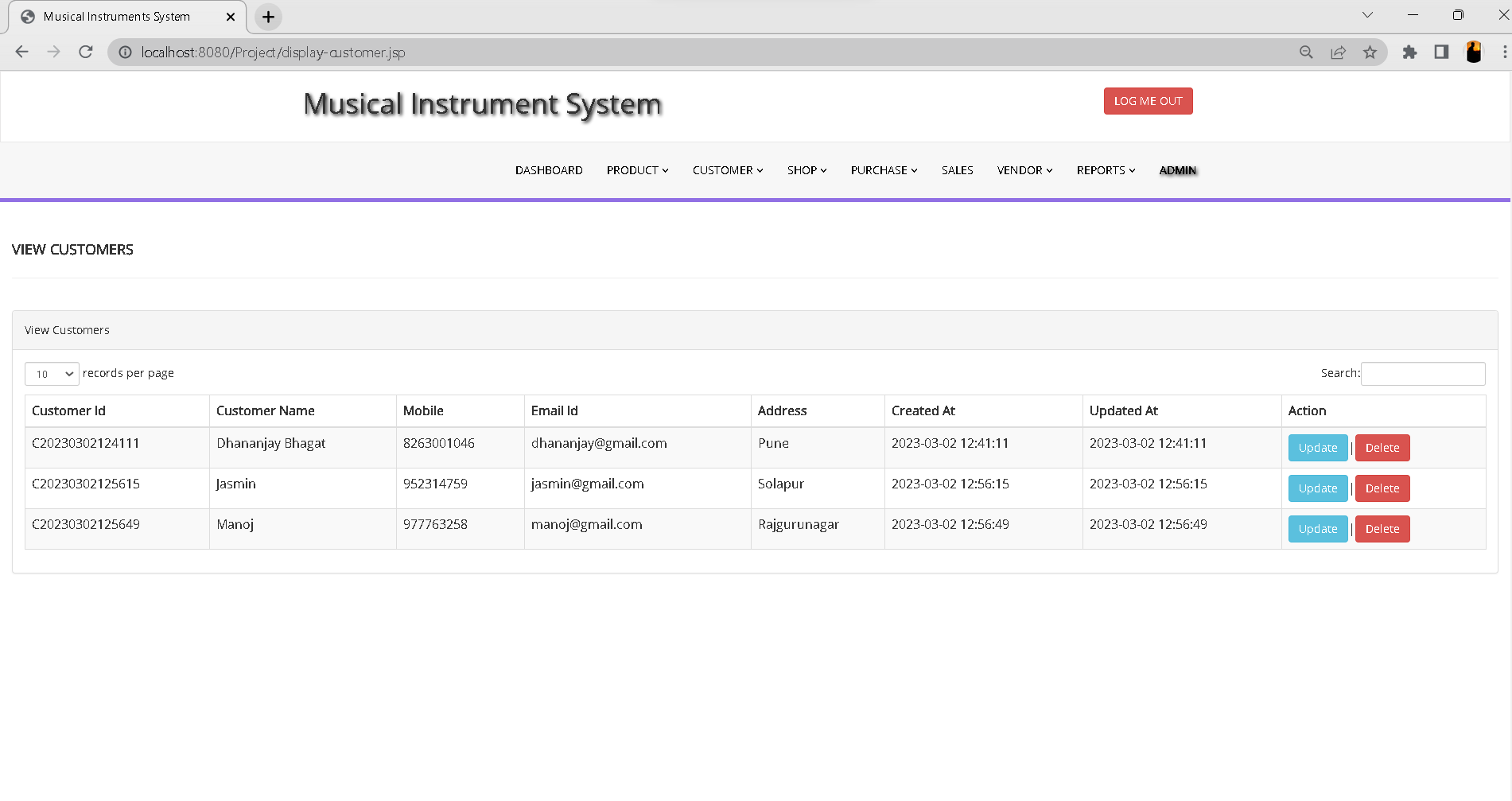
* **View Product**

****

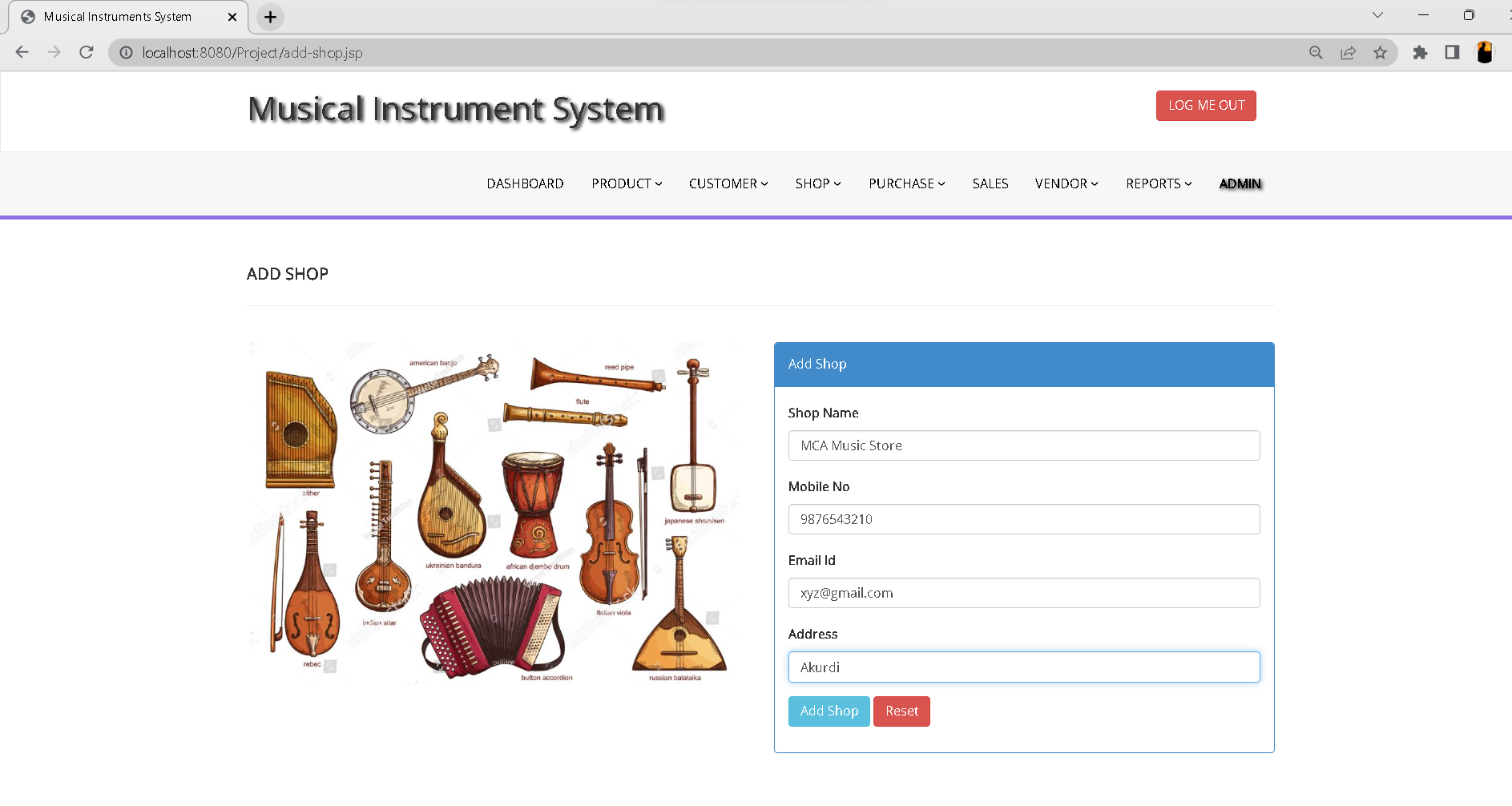
* **Add Customer**

****

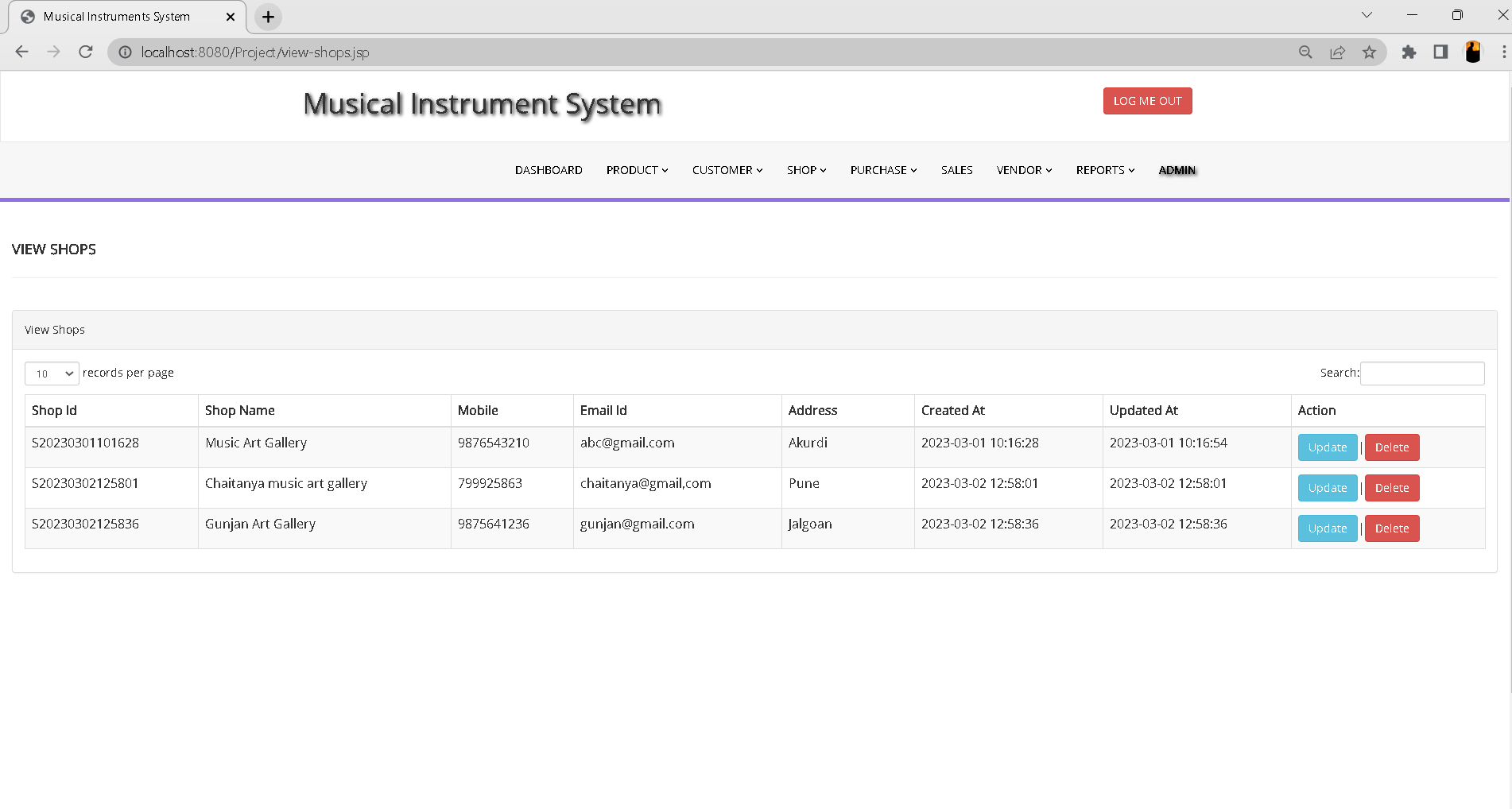
* **View Customer**

****

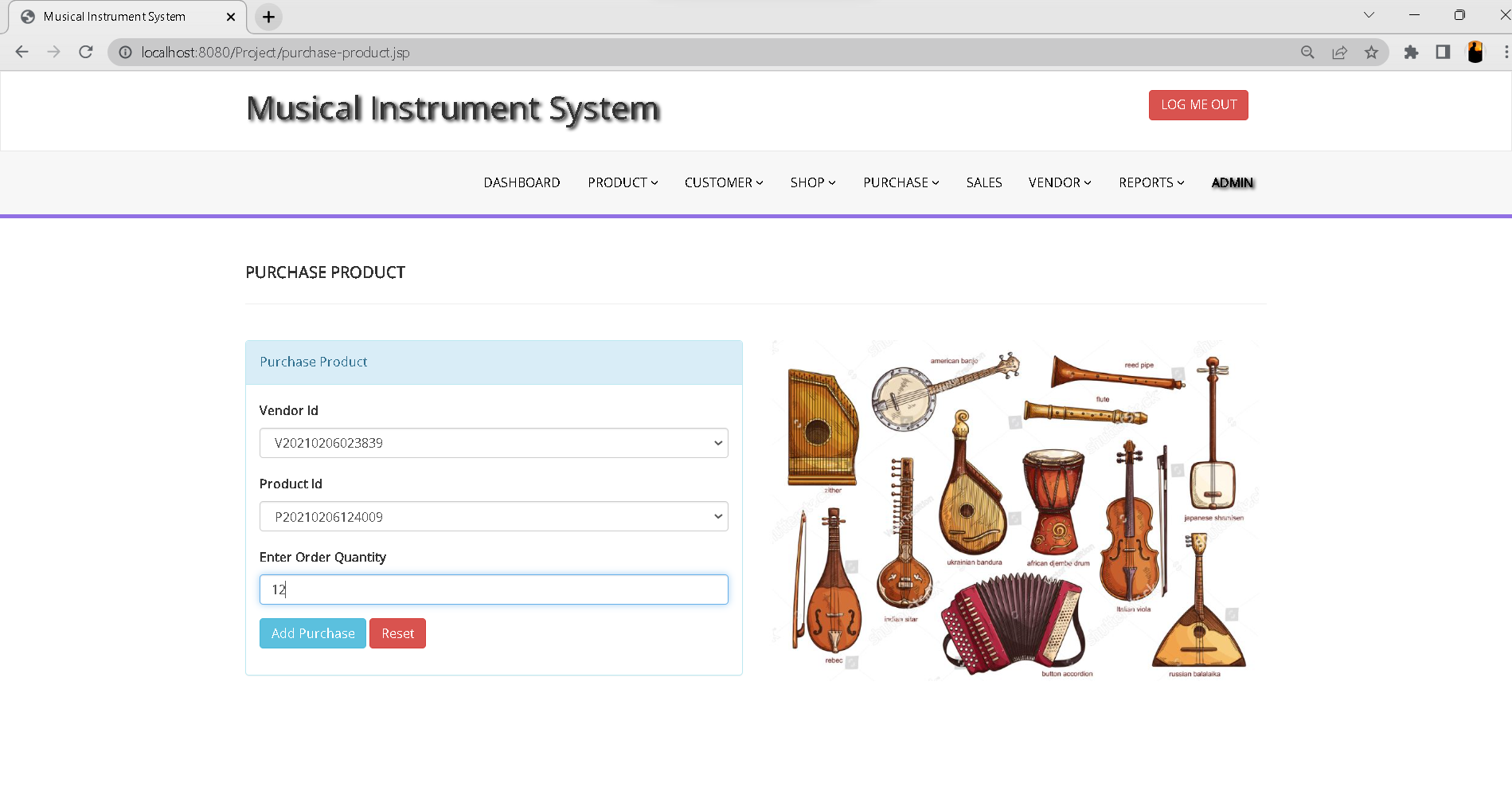
* **Add Shop**

****

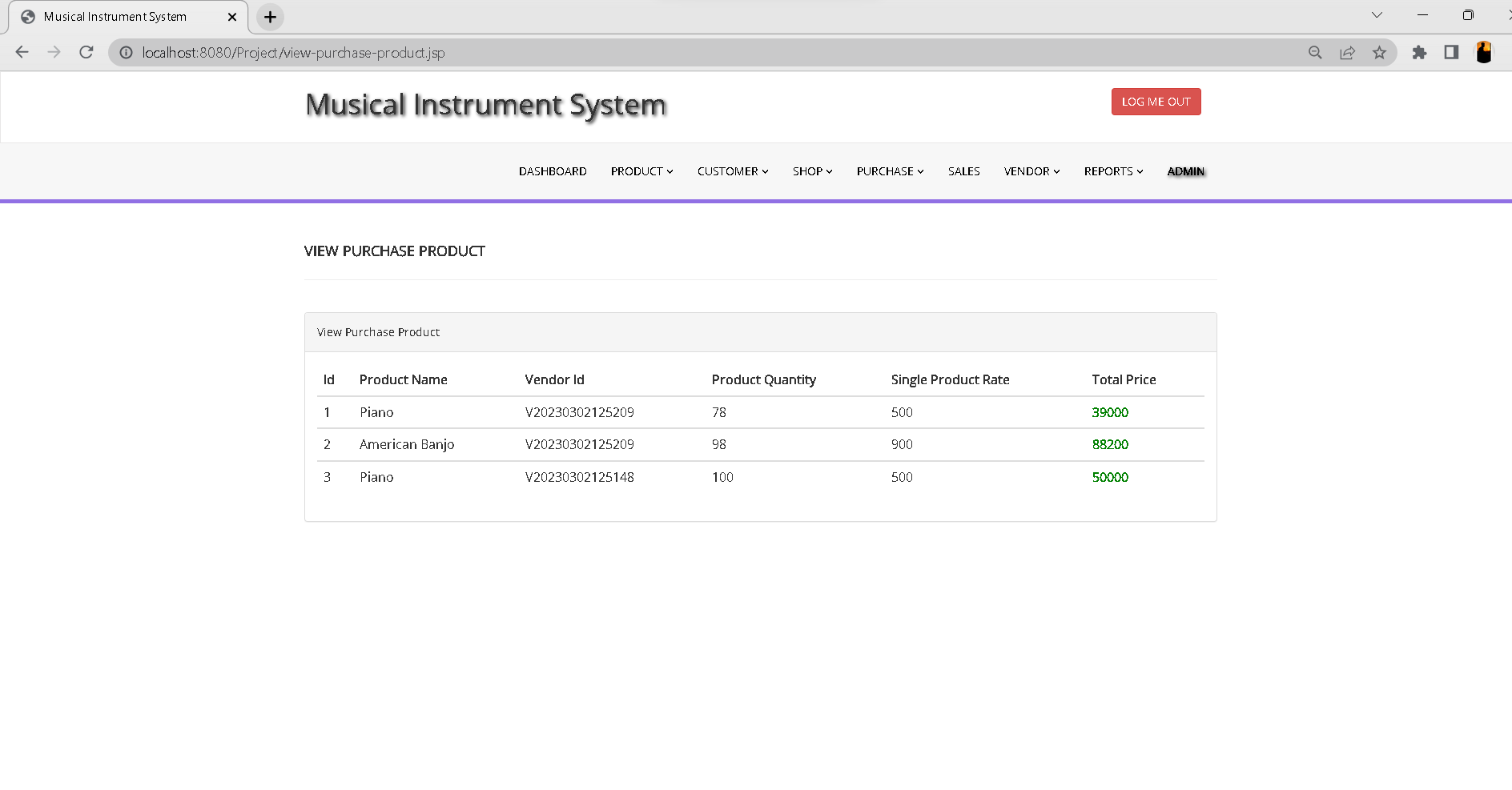
* **View Shop**

****

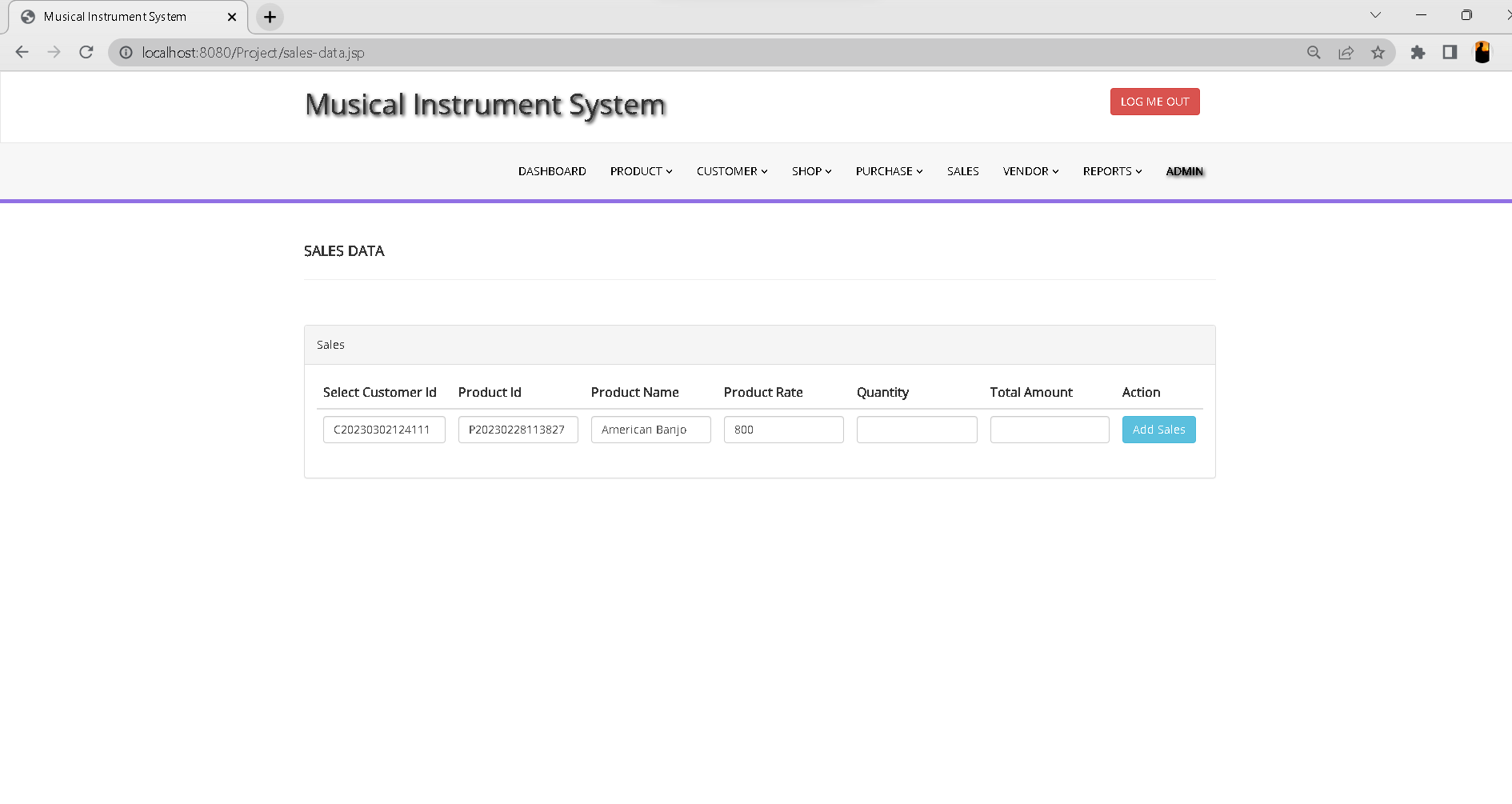
* **Purchase Product**

****

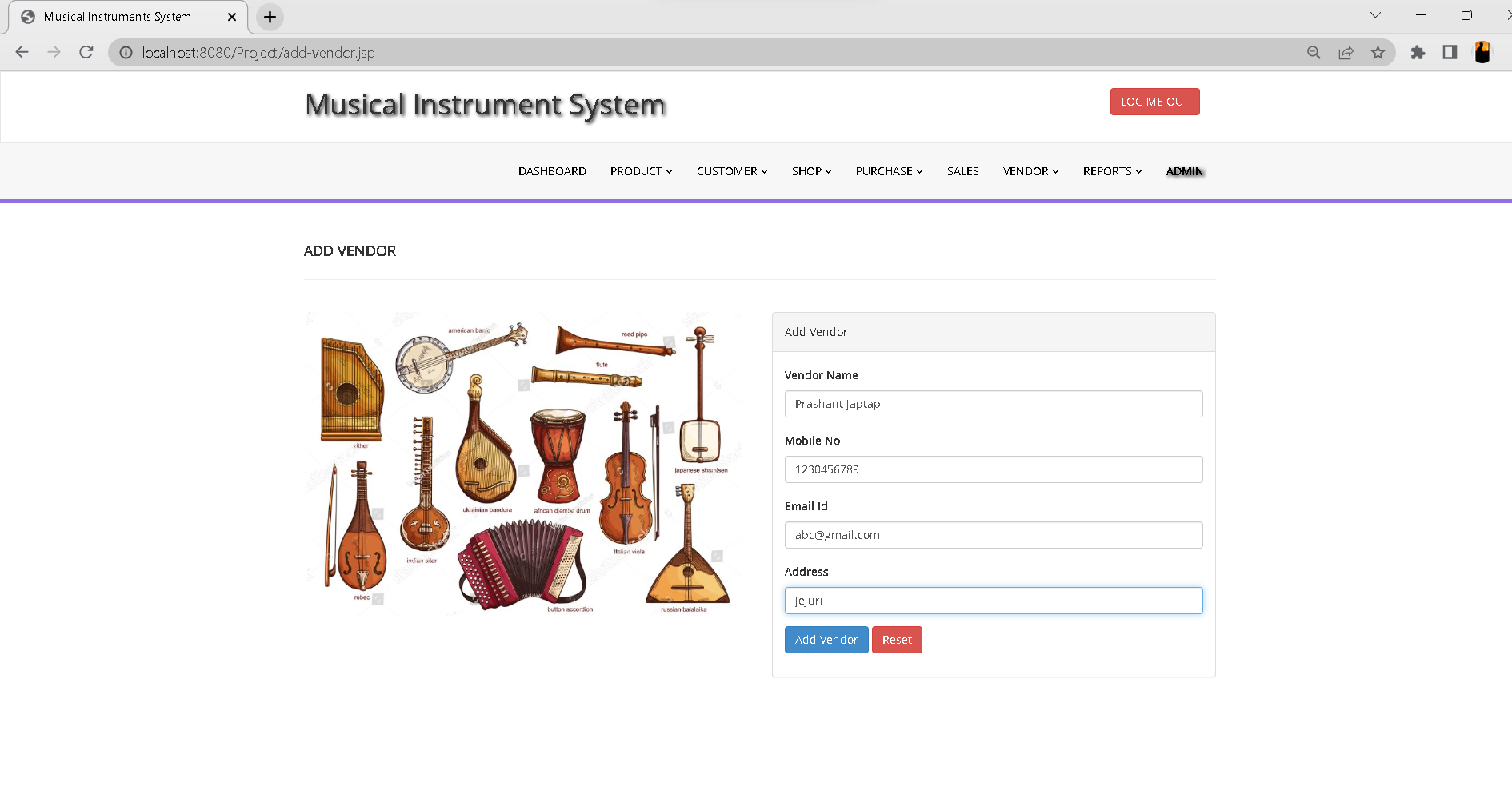
* **View Purchase Product**

****

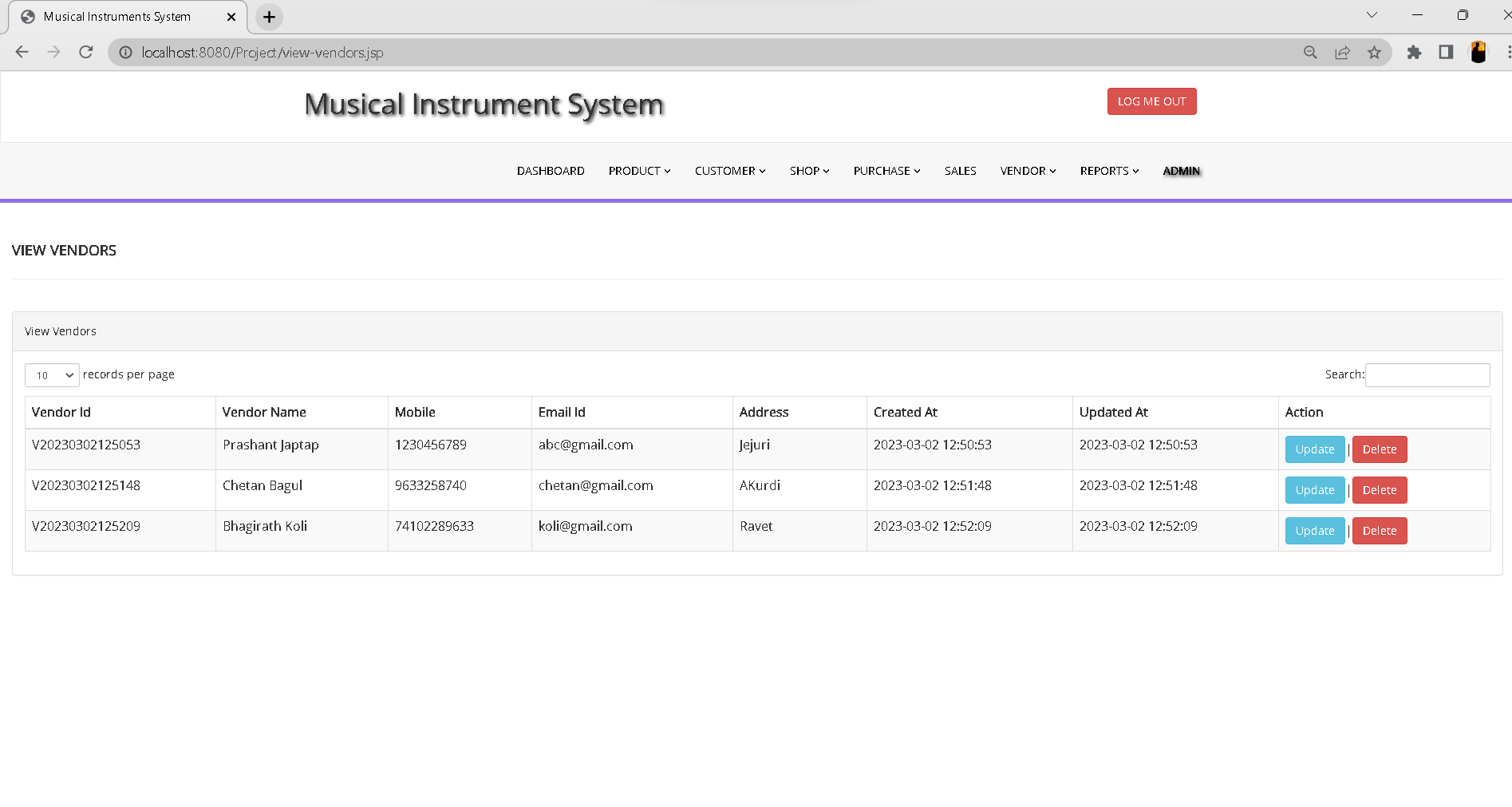
* **Sales Data**

****

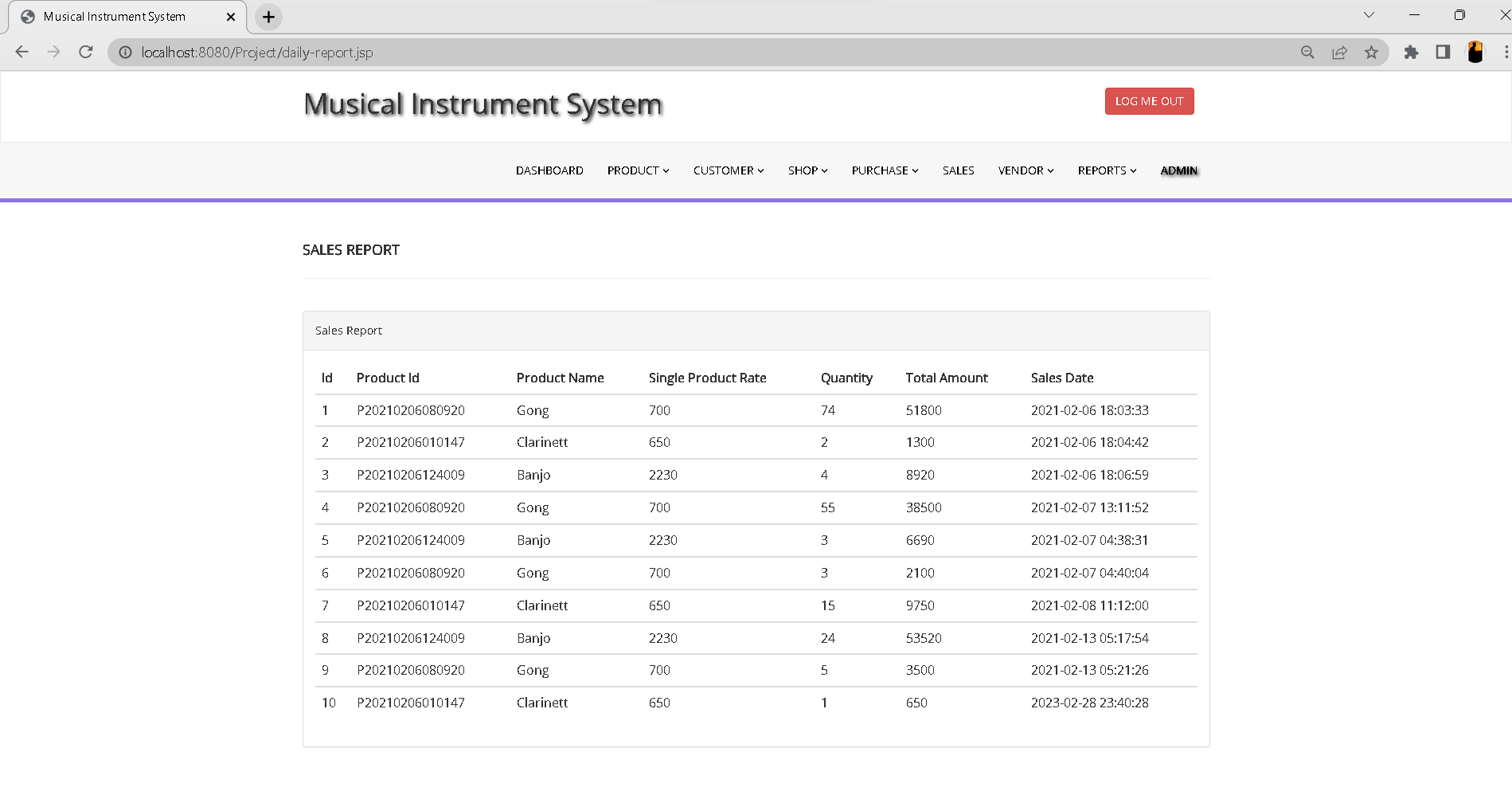
* **Add Vendor**

****

* **View Vendors**

****

* **Sales Report**

****

**2.8 Table Structure**

* **Admin Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Id | Integer | 100 | Primary Key | Admin Id |
| 2 | Uname | Varchar | 100 | Not Null | Admin Name |
| 3 | Password | Varchar | 100 | Not Null | Admin Password |
| 4 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |

* **Product Detail Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Prod\_id | Varchar | 100 | Primary key | Product Id |
| 2 | Prod\_name | Varchar | 45 | Not Null | Product Name |
| 3 | Prod\_qty | Integer | 45 | Not Null | Product Quantity |
| 4 | Prod\_rate | Integer | 45 | Not Null | Product Price |
| 5 | Vendor\_id | Varchar | 100 | Not Null | Vendor Id |
| 6 | Created\_at | Timestamp | 20 | Not Null | Current timestamp |
| 7 | Updated\_at | datetime | 8 | Not Null | Updated timestamp |

* **Customer Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | C\_id | Varchar | 100 | Primary Key | Customer id |
| 2 | C\_name | Varchar | 45 | Not Null | Customer Name |
| 3 | C\_contact | Varchar | 45 | Not Null | Customer Contact |
| 4 | C\_email | Varchar | 45 | Not Null | Customer Email id |
| 5 | C\_address | Varchar | 45 | Not Null | Customer Address |
| 6 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |
| 7 | Updated\_at | datetime | 8 | Not Null | Updated Timestamp |

* **Shop Details Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Shop\_id | Varchar | 100 | Primary Key | Shod id |
| 2 | Shop\_name | Varchar | 45 | Not Null | Shop Name |
| 3 | Contact | Varchar | 45 | Not Null | Shop Contact |
| 4 | Email | Varchar | 45 | Not Null | Shop Email id |
| 5 | Address | Varchar | 45 | Not Null | Shop Address |
| 6 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |
| 7 | Updated\_at | datetime | 8 | Not Null | Update Timestamp |

* **Vendor Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Vendor\_id | Varchar | 100 | Primary Key | Vendor id |
| 2 | Vendor\_name | Varchar | 45 | Not Null | Vendor Name |
| 3 | Contact | Varchar | 45 | Not Null | Vendor Contact |
| 4 | Email | Varchar | 45 | Not Null | Vendor Email id |
| 5 | Address | Varchar | 45 | Not Null | Vendor Address |
| 6 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |
| 7 | Updated\_At | datetime | 8 | Not Null | Updated Timestamp |

* **Purchase Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Purchase\_id | Varchar | 100 | Primary Key | Purchase id |
| 2 | Vendor\_id | Varchar | 100 | Not Null | Vendor id |
| 3 | Prod\_id | Varchar | 100 | Not Null | Product id |
| 4 | Order\_qty | Integer | 45 | Not Null | Order Quantity |
| 5 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |
| 6 | Updated\_at | datetime | 8 | Not Null | Updated Timestamp |

* **Sales Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr.No | Field Name | Data Type | Size | Constraints | Description |
| 1 | Sale\_id | Varchar | 100 | Primary Key | Sale id |
| 2 | cid | Varchar | 100 | Not Null | Customer id |
| 3 | Prod\_id | Varchar | 100 | Not Null | Product id |
| 4 | Prod\_name | Varchar | 45 | Not Null | Product Name |
| 5 | Quantity | Integer | 45 | Not Null | Quantity |
| 6 | Prod\_rate | Integer | 100 | Not Null | Product Price |
| 7 | Total\_amt | Integer | 100 | Not Null | Total Amount |
| 8 | Created\_at | Timestamp | 20 | Not Null | Current Timestamp |
| 9 | Updated\_at | datetime | 8 | Not Null | Updated Timestamp |

**3.Drawbacks And Limitations**

* Lack of Physical Interaction: One of the biggest limitations of an online musical instrument selling system is that customers cannot physically interact with the instruments before making a purchase. This can lead to dissatisfaction with the product, as customers may not be able to accurately judge the sound, feel, or quality of the instrument from online images and descriptions.
* Limited Accessibility: Online musical instrument selling systems may not be accessible to all potential customers, particularly those who do not have reliable internet access or who are not comfortable making purchases online.
* Security Concerns: Online transactions can be vulnerable to security breaches, which can lead to customer data being compromised or stolen. This can result in a loss of trust in the system and a decrease in sales.

**4.Proposed Enhancement**

* Current system is developed according to current requirements which can be added later. In this, system can be merged with another system to make bigger system invoking many functions on it.
* No project is ever complete in itself; there are always minor or major changes in the project according to user requirements.
* This project could be enhanced in the sense that it can overcome its limitations in the future as sample scope for enhancement.
* Latest electronic and software technologies can help to bring in more enhancement which would help to make the system more user friendly and also help to maintain adequate security.
* To make the application as online so that it would be helpful to everyone.

**5 . Conclusion**

* We have tried to develop a system that can be a great help to the modern technological world to register the requirements and needs from user.
* We have left all the options open so that if there is any other future requirement in the system by the user for enhancement of the system then it is possible to implement them.
* The MUSICAL INSTRUMENTS SYSTEM initiates the objective of providing an organizer with customized and powerful operating operations and management system side is built with all the options like adding customer, vendor, product, shop, with features for updating them whenever necessary, report generation, purchase details, sales details and many more.
* The interface provided is user friendly, flexible. We hope that the project will serve its purpose for which it is developed, by underlining success of the project.

**6 . Bibliography**

* **Website Reference**
* <https://www.w3schools.com>
* <https://www.javatpoint.com>
* <https://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html>
* <https://www.wikipedia.org/>
* [www.geeksforgeeks.org](http://www.geeksforgeeks.org)
* **Books Reference**
* Core Servlets and JSP, Vol 1 : Core Technologies
* HTML Black Book

**ANNEXURE 3 : SAMPLE PROGRAM CODE**

1. **Coding of Admin Login Page :**

**package** com.admin;

**import** java.io.IOException;

**import** java.sql.ResultSet;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.http.HttpSession;

**import** com.connection.DatabaseConnection;

@WebServlet("/AdminLogin")

**public** **class** AdminLogin **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

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

String uname=request.getParameter("uname");

String upass=request.getParameter("upass");

HttpSession hs=request.getSession();

**try** {

ResultSet resultset=DatabaseConnection.*getResultFromSqlQuery*("select \* from tbladmin where uname='"+uname+"' and password='"+upass+"'");

**if**(resultset.next()) {

hs.setAttribute("uname", resultset.getString("uname"));

response.sendRedirect("after-admin-login.jsp");

}**else** {

String message="Invalid credential, Please try again.";

hs.setAttribute("fail", message);

response.sendRedirect("index.jsp");

}

}**catch**(Exception e) {

e.printStackTrace();

}

}

}

1. **Coding for Customer Module :**

**Add Customer :**

**package** com.customer;

**import** java.io.IOException;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.http.HttpSession;

**import** com.connection.DatabaseConnection;

@WebServlet("/AddCustomer")

**public** **class** AddCustomer **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

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

String cid=DatabaseConnection.*generateCustomerId*();

String cname=request.getParameter("cname");

String mobile=request.getParameter("mobile");

String email=request.getParameter("email");;

String address=request.getParameter("address");

HttpSession hs=request.getSession();

**try** {

**int** addCustomer=DatabaseConnection.*insertUpdateFromSqlQuery*("insert into customer(c\_id,c\_name,c\_contact,c\_email,c\_address)values('"+cid+"','"+cname+"','"+mobile+"','"+email+"','"+address+"')");

**if**(addCustomer>0) {

String message="Customer add successfully.";

hs.setAttribute("customer-add", message);

response.sendRedirect("add-customer.jsp");

}

}**catch**(Exception e) {

e.printStackTrace();

}

}

}

**Delete Customer :**

**package** com.customer;

**import** java.io.IOException;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.http.HttpSession;

**import** com.connection.DatabaseConnection;

@WebServlet("/DeleteCustomer")

**public** **class** DeleteCustomer **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

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

String cid = request.getParameter("cid");

HttpSession hs=request.getSession();

**try** {

**int** deleteCustomer = DatabaseConnection.*insertUpdateFromSqlQuery*("delete from customer where c\_id='" + cid + "'");

**if** (deleteCustomer > 0) {

String message="Customer deleted successfully.";

hs.setAttribute("delete", message);

response.sendRedirect("display-customer.jsp");

} **else** {

response.sendRedirect("display-customer.jsp");

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

1. **Coding of Sales Module :**

**package** com.sales;

**import** java.io.IOException;

**import** java.sql.ResultSet;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.http.HttpSession;

**import** com.connection.DatabaseConnection;

@WebServlet("/CreateSales")

**public** **class** CreateSales **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

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

**int** productQuanity = 0;

**int** updateProductQuantity = 0;

**int** addSales = 0;

String salesId = DatabaseConnection.*generateSalesId*();

String custId = request.getParameter("custId");

String prod\_id = request.getParameter("prod\_id");

String pname = request.getParameter("pname");

String prate = request.getParameter("prate");

**int** pquantity = Integer.*parseInt*(request.getParameter("pquantity"));

String totalAmount = request.getParameter("totalAmount");

HttpSession hs = request.getSession();

**try** {

ResultSet resultset = DatabaseConnection.*getResultFromSqlQuery*("select prod\_qty from product\_details where prod\_id='" + prod\_id + "'");

**if** (resultset.next()) {

productQuanity = resultset.getInt("prod\_qty");

}

**if** (productQuanity > pquantity) {

addSales = DatabaseConnection.*insertUpdateFromSqlQuery*(

"insert into sales(sale\_id,cid,prod\_id,prod\_name,quantity,prod\_rate,total\_amt)values('"+ salesId + "','" + custId + "','" + prod\_id + "','" + pname + "','" + pquantity + "','"+ prate + "','" + totalAmount + "')");

updateProductQuantity = DatabaseConnection.*insertUpdateFromSqlQuery*("update product\_details set prod\_qty=prod\_qty-'" + pquantity+ "' where prod\_id='" + prod\_id + "'");

String message = "Sales data added";

hs.setAttribute("sale", message);

response.sendRedirect("sales.jsp");

} **else** {

String message = "Product quantity is not enough to purchase.";

hs.setAttribute("quantity-short", message);

response.sendRedirect("sales.jsp");

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

1. **Coding for Vendor Module :**

**Add Vendor :**

**package** com.vendor;

**import** java.io.IOException;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** jakarta.servlet.http.HttpSession;

**import** com.connection.DatabaseConnection;

@WebServlet("/AddVendor")

**public** **class** AddVendor **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

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

String vid=DatabaseConnection.*generateVendorId*();

String vname=request.getParameter("vname");

String mobile=request.getParameter("mobile");

String email=request.getParameter("email");;

String address=request.getParameter("address");

HttpSession hs=request.getSession();

**try** {

**int** addCustomer=DatabaseConnection.*insertUpdateFromSqlQuery*("insert into vendor(vendor\_id,vendor\_name,contact,email,address)values('"+vid+"','"+vname+"','"+mobile+"','"+email+"','"+address+"')");

**if**(addCustomer>0) {

String message="Vendor add successfully.";

hs.setAttribute("vendor-add", message);

response.sendRedirect("add-vendor.jsp");

}

}**catch**(Exception e) {

e.printStackTrace();

}

}

}

**Delete Vendor :**

package com.vendor;

import java.io.IOException;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import com.connection.DatabaseConnection;

@WebServlet("/DeleteVendor")

public class DeleteVendor extends HttpServlet {

private static final long serialVersionUID = 1L;

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

String vid = request.getParameter("vid");

HttpSession hs=request.getSession();

try {

int deleteVendor = DatabaseConnection.insertUpdateFromSqlQuery("delete from vendor where vendor\_id='" + vid + "'");

if (deleteVendor > 0) {

String message="Vendor deleted successfully.";

hs.setAttribute("delete", message);

response.sendRedirect("view-vendors.jsp");

} else {

response.sendRedirect("view-vendors.jsp");

}

} catch (Exception e) {

e.printStackTrace();

}

}

}