Online Shopping Platform

IT19341 – Introduction to OOPS and JAVA Project Report

Submitted by

DHARSHINI. C.K -231001034

ANANYA . S -231001013

BACHELOR OF TECHNOLOGY

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM (An Autonomous Institution)

RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

RAJALAKSHMI ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

Certified that this project titled "Movie Reservation System" is the bonafide work of **DHARSHINI .C.K(231001034), ANANYA .S(231001013)** who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr.P.Valarmathie	Mrs.Usha S
HEAD OF THE DEPARTMENT	COURSE INCHARGE
Information Technology Rajalakshmi Engineering College, Rajalakshmi Nagar, Thandalam Chennai – 602105	Information Technology Rajalakshmi Engineering College Rajalakshmi Nagar, Thandalam Chennai – 602105

INTERNAL EXAMINAR	**EXTERNAL EXAMINAR**

TABLE OF CONTENTS

1.1Abstract:

The primary objective of the JDBC-powered Online Shopping Platform in Java is to provide a reliable and efficient platform for users to seamlessly browse products, add them to their cart, and complete purchases, all while utilizing Java's database connectivity capabilities. This system ensures data integrity and consistency through JDBC transactions, establishing a secure connection with a relational database for real-time updates on product availability, order status, and user information. The user-friendly interface, coupled with scalability, makes it an effective solution for optimizing the online shopping experience, catering to various customer

.

1.2 Introduction:

The Online Shopping Platform is an innovative software solution designed to revolutionize tra ditio~~nal retail and e-commerce~~ experiences. This system automates the purchasing process, significantly reducing time and operational costs. Administered through a secure authentication system, the platform ensures that only authorized personnel can manage product listings, customer orders, inventory, and other critical functionalities. By integrating advanced technology into the online shopping experience, the platform provides a seamless, efficient, and user-friendly environment, offering a marked improvement over traditional manual retail methods and enhancing the overall customer journey.

1.3 Purpose:

Th e purpose of this project is to create an efficient and user-friendly online shopping platform that benefits both customers and administrators. The system aims to: -Simplify the process of browsing and purchasing products online. -Enhance the shopping experience by providing users with detailed product information, personalized recommendations, and easy checkout options. -Enable administrators to manage product listings, customer orders, inventory, and payment processing effectively. -Store and analyze customer and sales data for business insights and informed decision- making

1.4 Scope of the Project:

The envisioned Online Shopping Platform aims to seamlessly interact with administrators and efficiently fulfill all proposed functionalities. Built on Java (JDBC) with a MySQL database, the system ensures smooth management of product details, reducing response time for customer queries and providing real-time updates on inventory. This project addresses the complexities associated with manual shopping processes, storing comprehensive information about products, prices, and customer orders. Emphasizing features like verification, validation, security, and user-friendliness, the system strives to optimize the online shopping experience, offering a robust solution that encompasses the entire spectrum of e-commerce management.

1.5 Software Requirement Specification:

The Software Requirement Specification (SRS) outlines the functional and non-functional requirements of the Online Shopping Platform. This document serves as a blueprint for the development of the system, detailing the technologies, tools, and architecture required to build a robust, efficient, and user-friendly e-commerce solution. The Online Shopping Platform is designed to offer a seamless shopping experience to users while providing administrators with easy management capabilities. The front-end ensures that users can browse products, view their shopping carts, and complete transactions. The back-end handles complex business logic, integrates with the database for data storage, and manages user sessions and transactions. By combining robust technologies like Java, MySQL, and modern web development tools, the system aims to provide a scalable, secure, and responsive platform that meets the needs of both customers and administrators. This SRS ensures that the final system will be efficient, reliable, and easy to maintain.

Definitions, Acronyms, and Abbreviations:

SRS - Software Requirements Specification.

References and Acknowledgement:

[1] https://www.javatpoint.com/java-awt [2] https://www.javatpoint.com/java-swing

Overall Description:

The Online Shopping Platform provides authorized users with seamless access to product listings, customer orders, and transaction records, streamlining the online shopping process for customers worldwide. The system simplifies operations for e-commerce businesses, offering an efficient and user-friendly interface for browsing products , managing inventory, processing payments, and handling customer interactions, making it an ideal solution for modern retail environments.

Product Perspective:

Utilizing a client/server architecture, the system is designed to be compatible with the Microsoft Windows Operating System. The front end is developed using Java AWT and SWING, while the backend leverages the MySQL server for efficient data management.

Product Functionality:

The Online Shopping Platform provides user-friendly, menu-driven interfaces for: a) Admin Register: Registering new administrators. b) Admin Login: Logging in existing administrators. c) Add Product: Storing new product details, including descriptions, prices, and categories. d) View Product: Viewing and updating existing product information. e) Delete Product: Deleting existing products from the inventory. f) Add Order: Creating new orders for customers. g) Update Order: Viewing and modifying existing customer orders. h) Remove Admin: Deleting specific administrator accounts.

User and Characteristics:

Qualification: Users should have at least basic educational qualifications, such as matriculation, and be comfortable with English. Experience: Familiarity with the university registration process is advantageous. Technical Experience: Users are expected to have elementary knowledge of computers for optimal system interaction.

Operating Environment: *Hardware Requirements:* -
Processor: Any Processor over i3 - Operating System:
Windows 8, 10, 11 - Processor Speed: 2.0 GHz -
RAM: 4GB - Hard Disk: 500GB

Software Requirements:

- Database: MySQL - Frontend: Java (SWING, AWT) - Technology: Java (JDBC)

Constraints:

- System access limited to administrators. - Delete operation restricted to administrators without additional checks for simplicity. - Administrators must exercise caution during deletion to maintain data consistency.

Assumptions and Dependencies:

- System administrators create and confidentially communicate login IDs and passwords to users.

User Interface:

The Movie Reservation System provides user-friendly, menu-driven interfaces for:

a) Admin Register: Registering new administrators.
b) Admin Login: Logging in existing administrators.
c) Add Movie: Storing new movie details.
d) View Movie: Viewing and updating existing movie information.
e) Delete Movie: Deleting existing movies.
f) Add Reservation: Creating new reservations for users.
g) Update Reservation: Viewing and modifying existing reservations.
h) Remove Admin: Deleting specific administrator accounts.

Hardware Interface:

- Screen resolution of at least 640 x 480 or above. -
Compatible with any version of Windows 8, 10, 11.

Software Interface:

a) MS-Windows Operating System b) Java AWT and SWING for designing the front end c) MySQL for the backend d) Platform: Java Language e) Integrated Development Environment (IDE): Eclipse EE

Functional Requirements:

1. **Log in Module (LM):**
   - Users (administrators) access the Login Module. LM supports user login with a
   - username and password. Passwords are masked for security. Successful login
   - verification by the database administrator is required for access.
   -
2. **Registered Users Module (RUM):**
   - After successful login, users (administrators) can navigate through the application.
   - Users can view detailed information about products, orders, and inventory. Users can
   - update and maintain product details, including modifying inventory levels and prices.
   -
3. **Administrator Module (AM):**
   - Upon successful login, the system displays administrative functions. Functions include adding,
   - updating, and deleting product details. The "Add" function allows administrators to input new
   - product details, such as descriptions, prices, and categories. The "Update" function enables administrators to modify existing product details in the database. The "Delete" function allows
   - administrators to remove discontinued or unwanted products from the inventory. All add,
   - update, or delete requests trigger the AM module to communicate with the Server Module (SM)
   - for necessary database changes.
   -
   -
   -
4. **Server Module (SM):**
   - SM acts as an intermediary between various modules and the database (DB). Receives
   - requests from different modules and formats pages for display. Validates and executes
   - requests received from other modules. Handles communication with the database,
   - ensuring data consistency and integrity, especially regarding product details, orders,
   - and inventory management.

Non-Functional Requirements:

Performance:

- The system must handle real-time product browsing and order placement requests efficiently, ensuring a response time of less than 2 seconds for adding items to the cart
- and completing a purchase. Critical failures, such as payment processing errors, must
- be addressed instantly to ensure a smooth and uninterrupted shopping experience.
-
-

Reliability:

- The system is mission-critical; in case of abnormal operation or downtime, immediate
- measures should be taken to resolve the issue and restore normal functionality. The platform should ensure consistent and accurate performance, handling peak loads
- effectively, such as during flash sales or promotional events.
-

Availability:

- Under normal operating conditions, user requests for browsing products, viewing inventory , and processing orders should be handled within 2 seconds to maintain a seamless shopping
- experience. Immediate feedback on order status, payment confirmation, and delivery details
- should be provided to enhance user satisfaction.
-
-

Security:

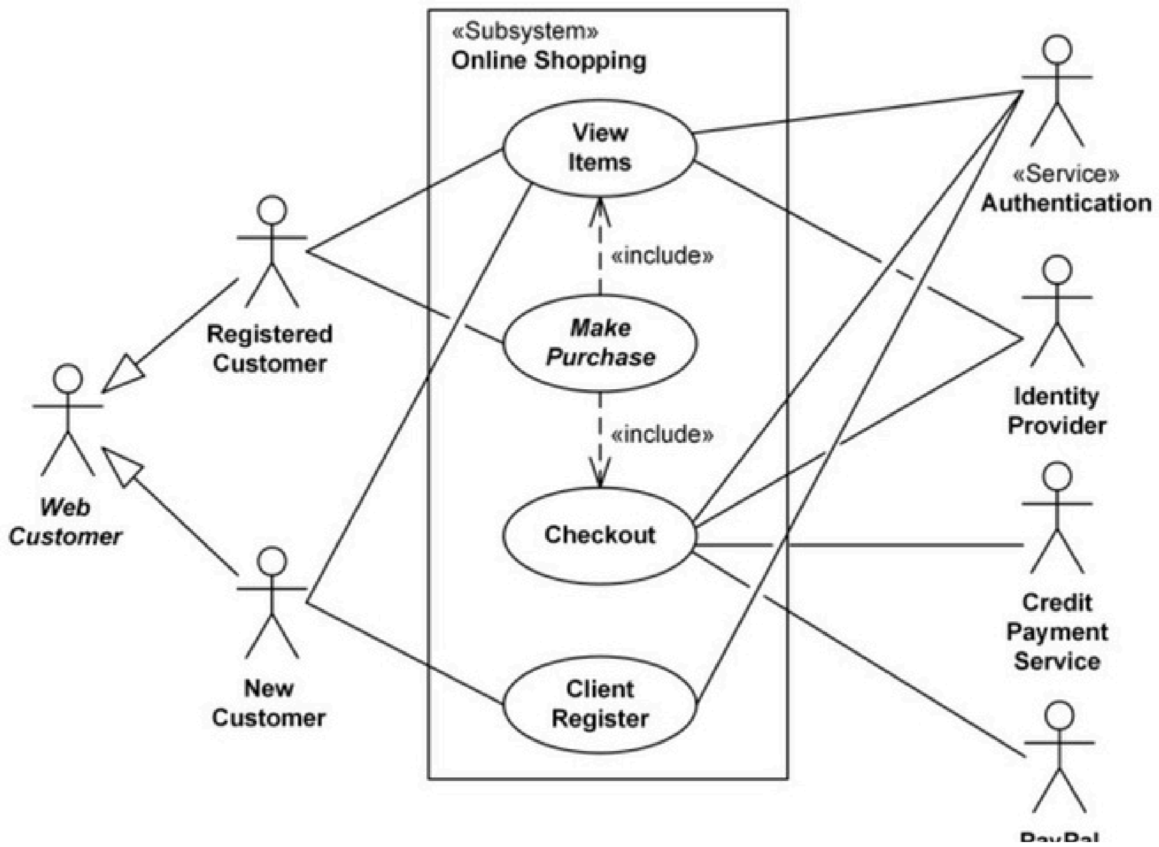
- A robust security mechanism must be implemented on the server side to prevent unauthorized
- access, protect user payment and personal information, and ensure the integrity of the
- shopping platform. User data, including account details, addresses, and transaction history,
- must be securely stored and managed to maintain confidentiality and comply with privacy
- regulations.

Maintainability:

- Comprehensive design documents outlining software and database maintenance procedures
- must be created to facilitate regular updates and modifications to the online shopping platform. Administrative access should be provided for maintaining the platform at both the
- front end and back end, ensuring long-term functionality, scalability, and adaptability to
- evolving business needs.

4o

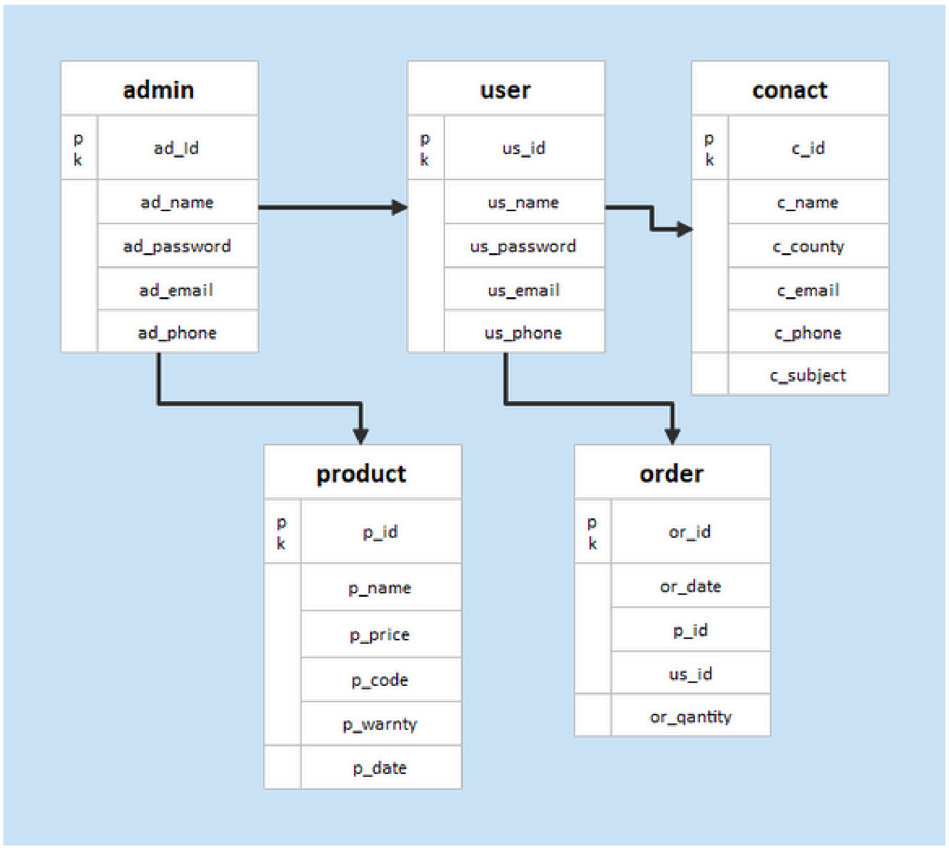
2.System Flow Diagrams: 2.1.Use Case Diagrams :



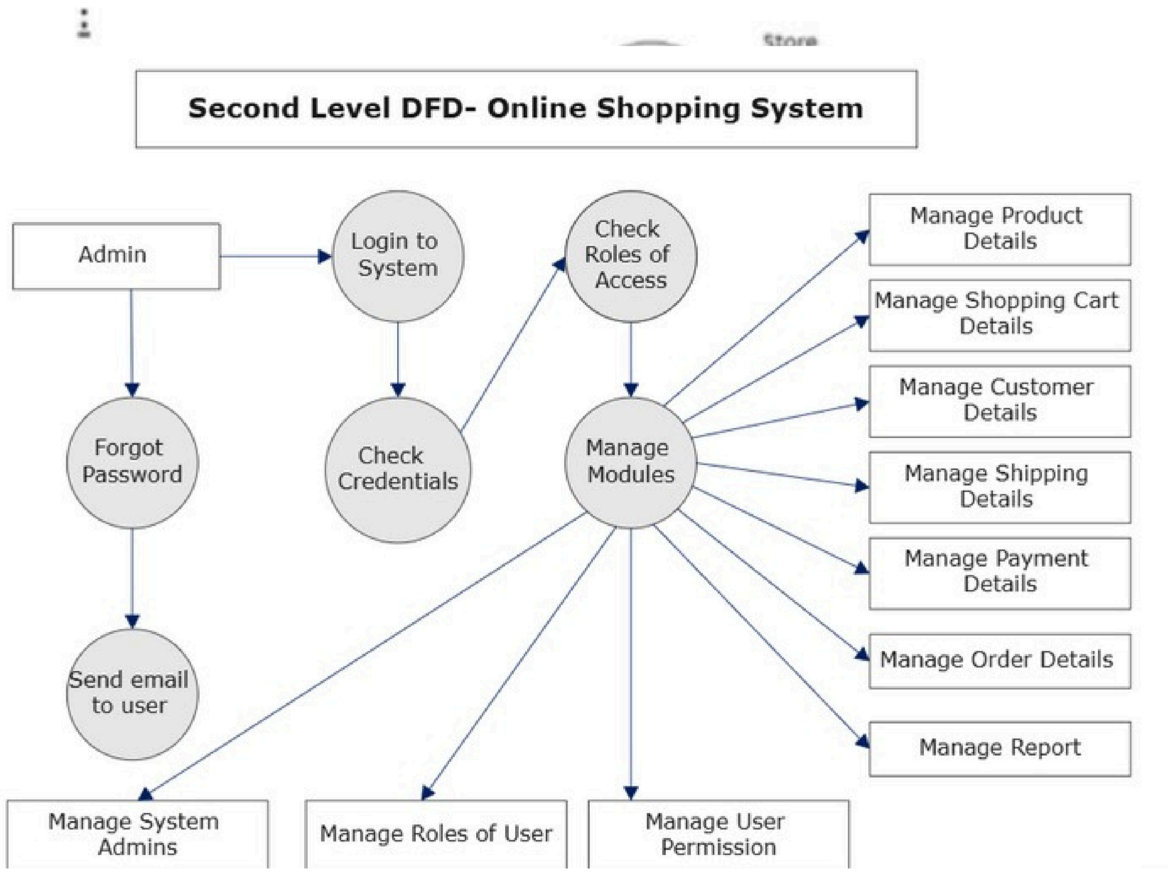
2.2 Entity-relationship diagram:

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

Entity -relationship diagram:



2.3 Data-flow diagram:



3. Module description

Admin: Register:

- Admin can register with a username and password to gain access to the platform's administrative functions.

Login:

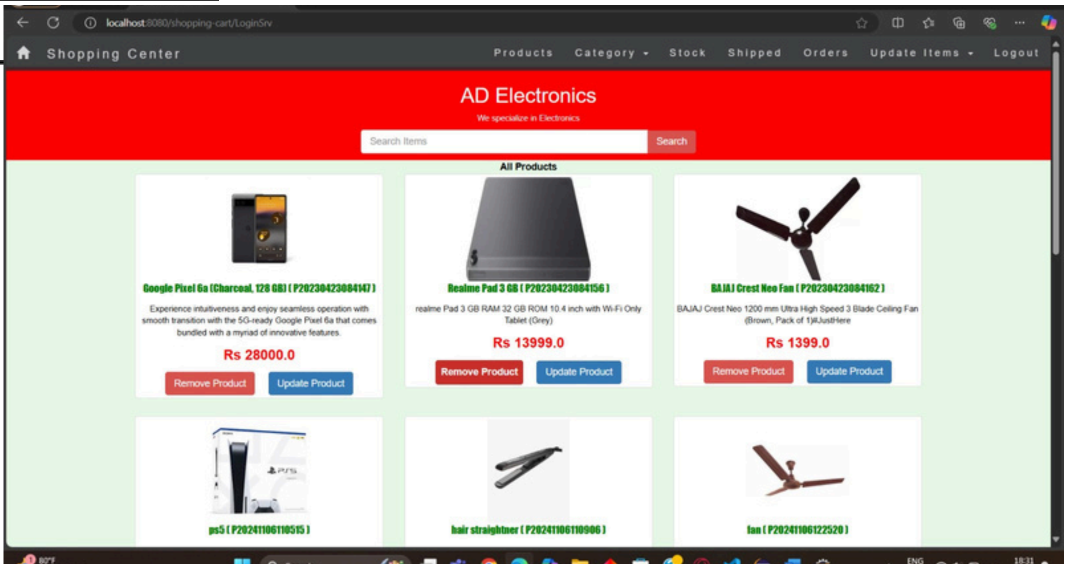
- Admin can log in using their registered username and password.

After Login:

1. **Add Product:**
    - Admin can add new products to the platform, including details such as product name, category, price, stock quantity, and description.
2. **View Product:**
    - Admin can view and update product details such as the name, price, stock levels, or description to keep the inventory accurate.
3. **Delete Product:**
    - Admin can remove outdated or discontinued products from the system.
4. **Add Order:**
    - Admin can manually add orders for customers, including details such as product selection, quantity, and payment status.
5. **Update Order:**
    - Admin can modify existing order details, such as updating the shipping address, product quantity, or order status.
6. **Remove Admin:**
    - Admin details can be deleted if necessary, ensuring proper user management and security compliance.

This structure ensures efficient management of the platform and streamlines both customer and administrator interactions.

4.Implementation: 4.1 Design: Home Page:


localhost:8080/shopping-cart/LoginSrv

Shopping Center — Products, Category, Stock, Shipped, Orders, Update Items, Logout

AD Electronics

We specialize in Electronics

All Products

Google Pixel 6a (Charcoal, 128 GB) (P20230423084147)
Experience intuitiveness and enjoy seamless operation with smooth transition with the 5G-ready Google Pixel 6a that comes bundled with a myriad of innovative features.
Rs 28000.0

Realme Pad 3 GB (P20230423084156)
realme Pad 3 GB RAM 32 GB ROM 10.4 inch with Wi-Fi Only Tablet (Grey)
Rs 13999.0

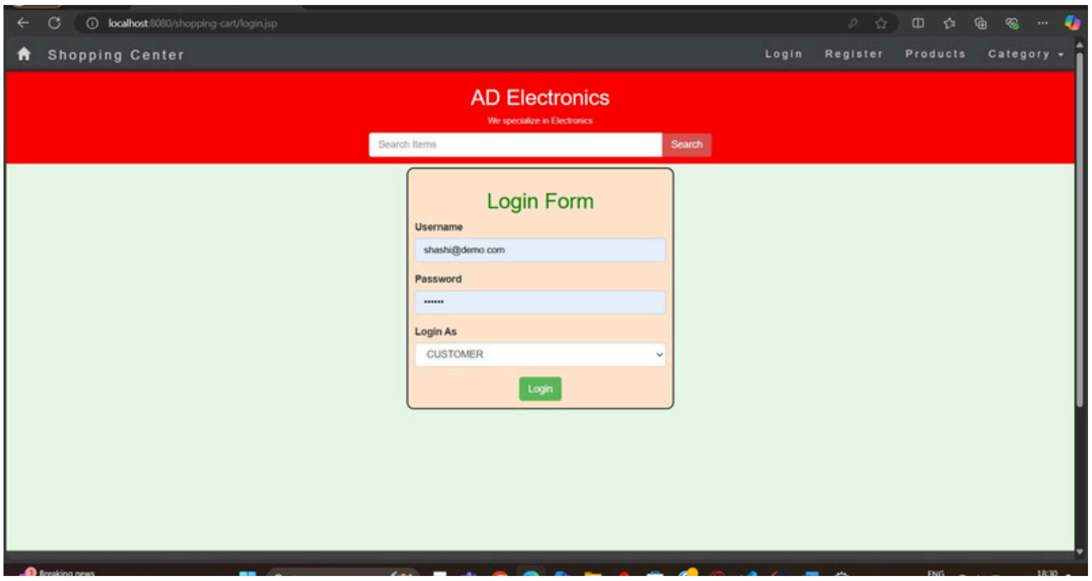
BAJAJ Crest Neo Fan (P20230423084162)
BAJAJ Crest Neo 1200 mm Ultra High Speed 3 Blade Ceiling Fan (Brown, Pack of 1)#JustHere
Rs 1399.0

ps5 (P20241106110515)

hair straightner (P20241106110906)

fan (P20241106122520)

Login page:


localhost:8080/shopping-cart/login.jsp

Shopping Center — Login, Register, Products, Category

AD Electronics

We specialize in Electronics

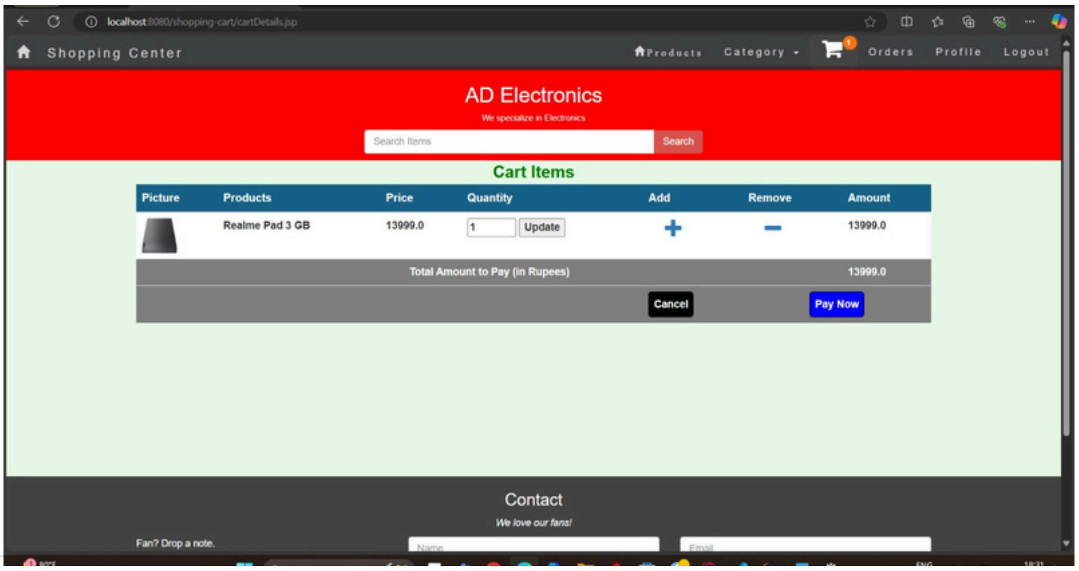
Login Form

Username: shashi@demo.com

Password: ••••••

Login As: CUSTOMER

Cart Items


localhost:8080/shopping-cart/cartDetails.jsp

Shopping Center — Products, Category, Orders, Profile, Logout

AD Electronics

We specialize in Electronics

Cart Items

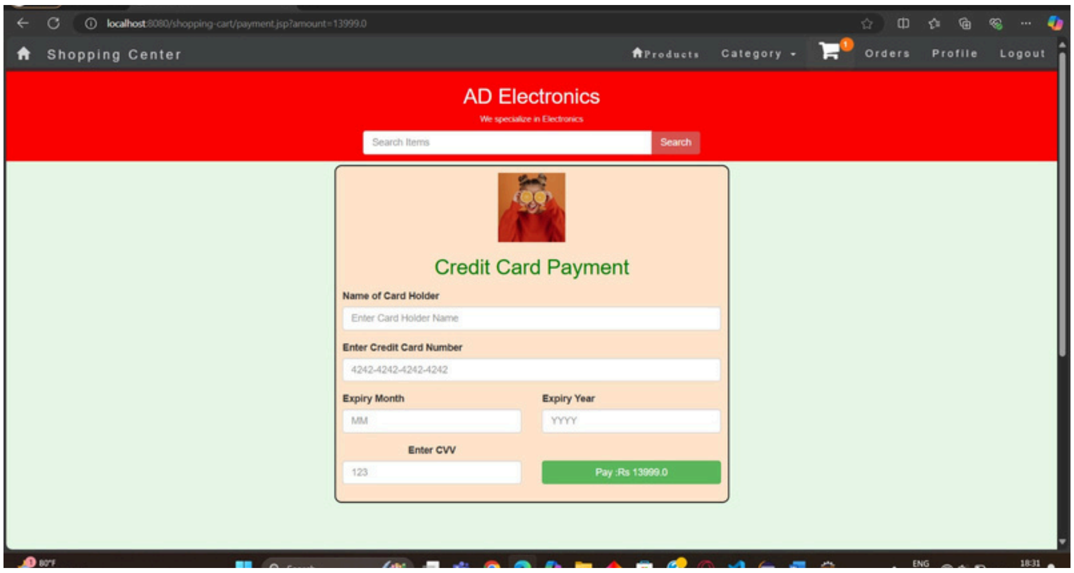
Picture	Products	Price	Quantity	Add	Remove	Amount
	Realme Pad 3 GB	13999.0	1 Update	+	−	13999.0
	Total Amount to Pay (in Rupees)					13999.0

Contact

We love our fans!

Fan? Drop a note.

Credit Card Payment


localhost:8080/shopping-cart/payment.jsp?amount=13999.0
Shopping Center
Products
Category
Orders
Profile
Logout
AD Electronics
We specialize in Electronics
Search Items
Search

Credit Card Payment

Name of Card Holder: Enter Card Holder Name

Enter Credit Card Number: 4242-4242-4242-4242

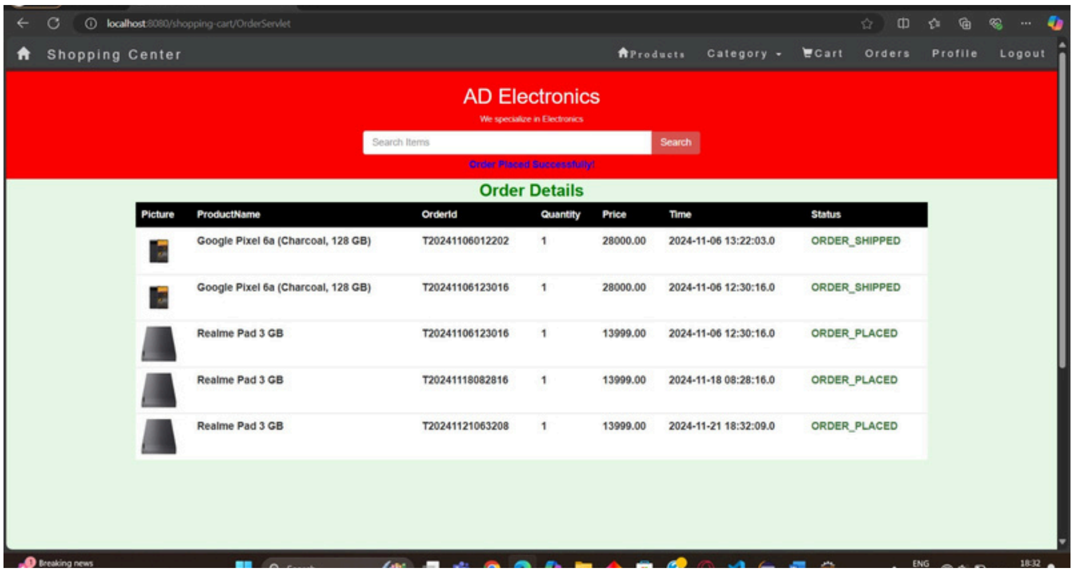
Expiry Month: MM

Expiry Year: YYYY

Enter CVV: 123

Pay :Rs 13999.0

Order Details & Status



Order Placed Successfully!

Order Details

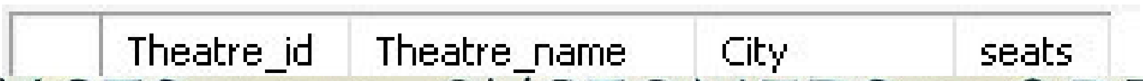
Picture	ProductName	OrderId	Quantity	Price	Time	Status
	Google Pixel 6a (Charcoal, 128 GB)	T20241106012202	1	28000.00	2024-11-06 13:22:03.0	ORDER_SHIPPED
	Google Pixel 6a (Charcoal, 128 GB)	T20241106123016	1	28000.00	2024-11-06 12:30:16.0	ORDER_SHIPPED
	Realme Pad 3 GB	T20241106123016	1	13999.00	2024-11-06 12:30:16.0	ORDER_PLACED
	Realme Pad 3 GB	T20241118082816	1	13999.00	2024-11-18 08:28:16.0	ORDER_PLACED
	Realme Pad 3 GB	T20241121063208	1	13999.00	2024-11-21 18:32:09.0	ORDER_PLACED

4.2 Database Design:

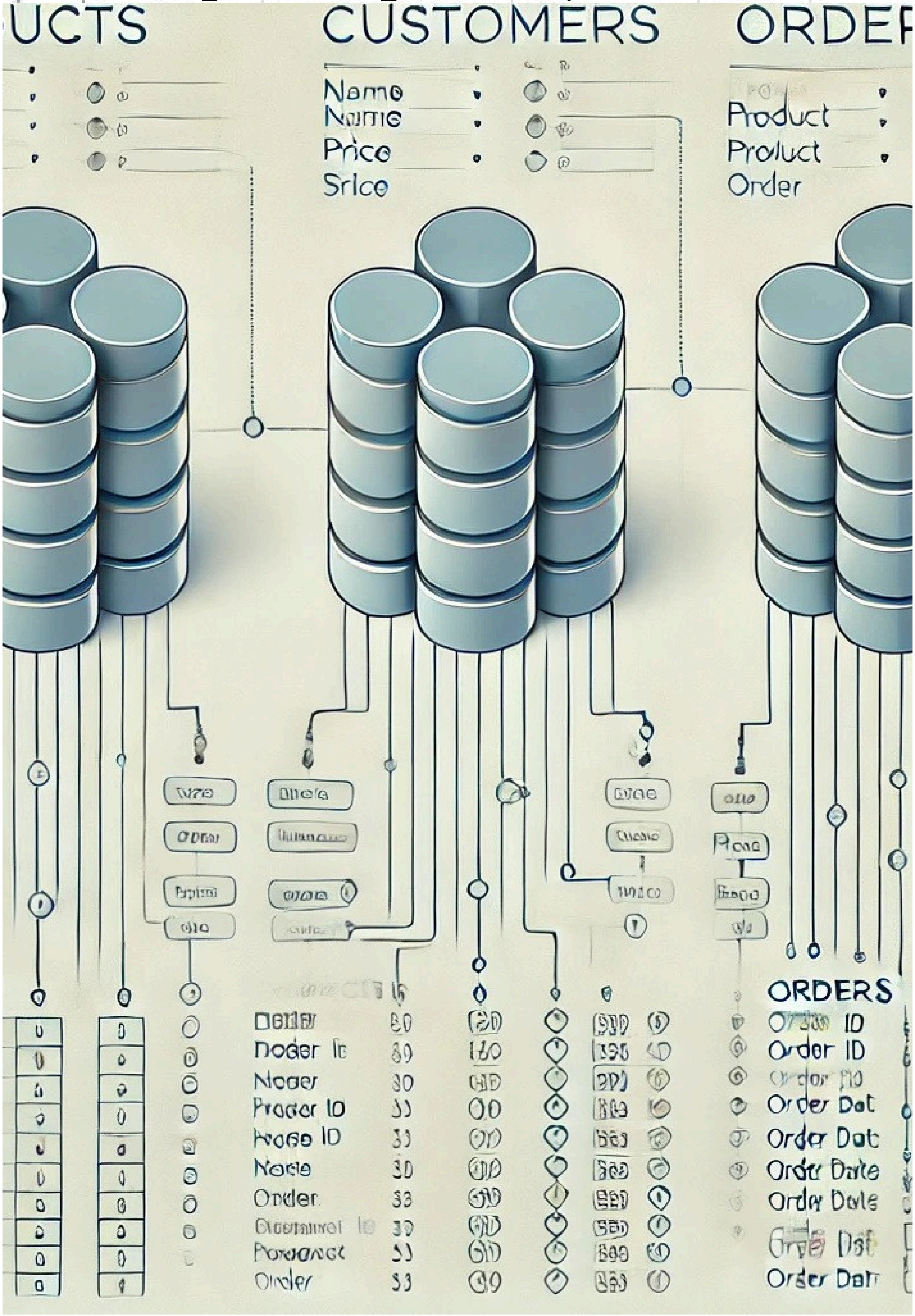
The data in the Online Shopping Platform system is stored and retrieved from a relational database. Designing the database is an essential part of the system design process. Data elements and data structures identified during the analysis stage are organized and structured to design the storage and retrieval system effectively. A database is a collection of interrelated data stored with minimal redundancy to serve many users efficiently. The objective is to provide fast, reliable, and flexible access to data for users while keeping storage costs low. Relationships between data entities are carefully established, and unnecessary data is removed. Normalization is applied to achieve internal consistency of data, minimize redundancy, and ensure maximum stability. This process reduces the storage required, minimizes the chances of data inconsistencies, and optimizes the performance of data updates and retrieval. For the Online Shopping Platform, the MySQL database has been chosen to develop and manage relevant databases. This ensures efficient handling of critical data, such as product details, customer information, order transactions, and inventory levels. The database design als o supports scalability and reliability to accommodate the platform's growth and evolving requirements.

Online Shopping Platform

MySQL tables :



	Theatre_id	Theatre_name	City	seats



4.3 Code:

```
package com.DA.service.impl;

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

import com.shashi.beans.UserBean;
import com.shashi.constants.IUserConstants;
import com.shashi.service.UserService;
import com.shashi.utility.DBUtil;
import com.shashi.utility.MailMessage;

public class UserServiceImpl implements UserService {

@Override
public String registerUser(String userName, Long mobileNo, String emailId, String address, int pinCode,
String password) {

UserBean user = new UserBean(userName, mobileNo, emailId, address, pinCode, password);

String status = registerUser(user);

return status;
}

@Override
public String registerUser(UserBean user) {

String status = "User Registration Failed!";

boolean isRegtd = isRegistered(user.getEmail());

if (isRegtd) {
status = "Email Id Already Registered!";
return status;
}
Connection conn = DBUtil.provideConnection();
PreparedStatement ps = null;
if (conn != null) {
System.out.println("Connected
        Successfully!");
}
```

```
try

{
ps = conn.prepareStatement("insert into " + IUserConstants.TABLE_USER + " values(?,?,?,?,?,?)");

ps.setString(1, user.getEmail());
ps.setString(2, user.getName());
ps.setLong(3, user.getMobile());
ps.setString(4, user.getAddress());
ps.setInt(5, user.getPinCode());
ps.setString(6, user.getPassword());

int k = ps.executeUpdate();

if (k > 0) {
status = "User Registered Successfully!";
MailMessage.registrationSuccess(user.getEmail(), user.getName().split(" ")[0]);
}

} catch (SQLException e) {
status = "Error: " + e.getMessage();
e.printStackTrace();
}

DBUtil.closeConnection(ps);
DBUtil.closeConnection(ps);

return status;
}

@Override
public boolean isRegistered(String emailId) {
boolean flag = false;

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;
ResultSet rs = null;

try {
ps = con.prepareStatement("select * from user where email=?");

ps.setString(1, emailId);
if (rs.next())
flag = true;

} catch (SQLException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

DBUtil.closeConnection(con);
DBUtil.closeConnection(ps);
DBUtil.closeConnection(rs);

return flag;
}

@Override
public String isValidCredential(String emailId, String password) {
String status = "Login Denied! Incorrect Username or Password";

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;
ResultSet rs = null;

try

{
ps = con.prepareStatement("select * from user where email=? and password=?");

ps.setString(1, emailId);
ps.setString(2, password);

rs = ps.executeQuery();

if (rs.next())
status = "valid";
} catch (SQLException e) {
status = "Error: " + e.getMessage();
e.printStackTrace();
}

DBUtil.closeConnection(con);
return status;
}
```

```
@Override public UserBean getUserDetails(String emailId, String
password) {

User Bean user = null;

Connection con = DBUtil.provideConnection();

PreparedStatement ps = null;
ResultSet rs = null;

try {
ps = con.prepareStatement("select * from user where email=? and password=?");
ps.setString(1, emailId);
ps.setString(2, password);
rs = ps.executeQuery();

if (rs.next()) {
user = new UserBean();
user.setName(rs.getString("name"));
user.setMobile(rs.getLong("mobile"));
user.setEmail(rs.getString("email"));
user.setAddress(rs.getString("address"));
user.setPinCode(rs.getInt("pincode"));
user.setPassword(rs.getString("password"));

return user;
}

} catch (SQLException e)

{
e.printStackTrace();
}
DBUtil.closeConnection(con);
DBUtil.closeConnection(ps);
DBUtil.closeConnection(rs);

return user;
}

@Override
public String getFName(String emailId) {
String fname = "";

Connection con = DBUtil.provideConnection();
```

```
try {
ps = con.prepareStatement("select name from user where email=?");
ps.setString(1, emailId);

rs = ps.executeQuery();

if (rs.next()) {
fname = rs.getString(1);

fname = fname.split(" ")[0];

}

} catch (SQLException e) {

e.printStackTrace();
}

return fname;
}

@Override
public String getUserAddr(String userId) {
String userAddr = "";

Connection con = DBUtil.provideConnection();
PreparedStatement ps = null;
ResultSet rs = null;

try {
ps = con.prepareStatement("select address from user where email=?");

ps.setString(1, userId);

rs = ps.executeQuery();

if (rs.next())
userAddr = rs.getString(1);

} catch (SQLException e) {

e.printStackTrace();
}
```

```
return userAddr;
}
```

Conclusion:

The Movie Reservation System project, executed under experienced guidance, embodies a meticulous approach to design and implementation. With a focus on user-friendly functionalities like adding/viewing movies and reservation management, the system ensures a seamless movie-going experience. Robust security measures, particularly in the "Remove Admin" module, highlight a commitment to data integrity and system protection. The project stands as a well-rounded solution, meeting current requirements while allowing for future adaptability and enhancements.

Reference links: [1]

https://www.javatpoint.com/java-awt [2]
https://www.javatpoint.com/java-swing