

CORE JAVA PROJECT

PROJECT TITLE: ONLINE BOOK STORE PROJECT

Project Report on

Online Book Store

Submitted for the partial fulfillment of the

CERTIFIED JAVA FULL-STACK PROFESSIONAL COURSE

Conducted by Edubridge

Submitted by S.Deepika

Enrollment Number: EBEON0122553061

Batch:2021-6769

Under the Guidance of

Trainer Mrs.Indrakka Malli.,

ABSTRACT

The main objective of the project is to create an online book store that allows users to view and purchase a book online based on name, author, and category. The selected books are displayed in a tabular format and the user can order their books online through credit card payment. Using this Website, the user can purchase a book online instead of going out to a book store and wasting time. There are many online book stores like Powell's, and Amazon which were designed using Html. I want to develop a similar website using eclipse, SQL Server. An Online Book store is an online web application where the customer can purchase books online. Through a web browser, the customers can view a book by its name or author and finally purchase using a credit card transaction. The user can log in using their account details or new customers can set up an account very quickly by registration. They should give the details of their name, contact number, and shipping address when they are ready to order. The Online Book Store Website provides customers with online shopping through a web browser. A customer can create, sign in to his account, order items and purchase using his credit card details. The Administrator will have additional functionalities when compared to the common user. He can view, add, delete and update the book details, and book categories. This application is developed using java and SQL programming language.

INTRODUCTION

1.1 Purpose and Motivation

Online book store project in java developed mainly for bookstores and shops for the book purchasing process. This application will allow a user to buy or get the book online in just some simple steps. The user needs to have an internet connection and browser to access the internet. Let's check the functionality and role of the users in the online book store step by step.

MOTIVATION:

The motivation to create this project has many sources

- Interest to develop a good user-friendly website with many online transactions using a database.
- To increase my knowledge horizon in technologies like JAVA, and SQL.
- To gain good experience in JAVA before joining a full-time job.
- To gain good practice in coding.

Features and functionality of the online book store

Two main application actors will interact directly with the application: Admin and a user. Admin can add, edit and delete the book in the shop and the user can view, purchase the required book and also cancel the ordered book.

DESIGN AND METHODOLOGY:

In this technique several technical approaches are used:

Operating System: Windows 11

Software Tool: IDE, MyEclipse, Eclipse

Programming Language: Java as a Programming language

Database: SQL, Structured Query Language for relational databases and is used to access and manipulate databases.

Java JDBC technology for database connection

Server: Apache Tomcat

IMPLEMENTATION:

INITIAL SETUP:

(IN JAVA):

Maven Project Name: ONLINE BOOK STORE Project.

Package Name: com.onlinebookstore

Java Classes: 1. BookStoreMain

2.DbConnect

3.BookStoreProcess

(IN DATABASE)

Database Creation:

Creating a database using SQL Query

```
mysql> create database OnlineBookStore;
```

Database created...

Query to use the created Database.

```
mysql> use OnlineBookStore;  
Database changed
```

I.DATABASE OVERVIEW:

The online bookstore database had created by using MYSQL. It is made up of six tables:

- admin_registration
- user_registration
- bookstore
- customerorderdetails
- deliverydetail
- customersaccount
- AdminsAccount

Table Creation:

```
+-----+
| Tables_in_onlinebookstore |
+-----+
| admin_registration |
| adminaccount       |
| bookstore          |
| customerorderdetails |
| customersaccount   |
| deliverydetail     |
| user_registration  |
+-----+
7 rows in set (0.01 sec)
```

ADMIN_REGISTRATION TABLE:

Define the registration process of ADMIN.

Admin Registration Table Structure:

```
mysql> describe admin_registration;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emailid    | varchar(30)   | NO   | PRI | NULL     |       |
| Username   | varchar(30)   | YES  |     | NULL     |       |
| Password   | varchar(30)   | NO   |     | NULL     |       |
| repass     | varchar(20)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

USER_REGISTRATION TABLE:

Define the registration process of USER.

User Reservation Table Structure:

```
mysql> describe user_registration;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emailid    | varchar(30)   | NO   | PRI | NULL    |       |
| Username   | varchar(30)   | YES  |     | NULL    |       |
| Password   | varchar(30)   | NO   |     | NULL    |       |
| repass     | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

BOOKSTORE TABLE:

Define the list of book and book details processes which are available in the bookstore.

Book Store Table Structure:

```
mysql> describe BookStore;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| BookId     | int           | NO   | PRI | NULL    |       |
| Category   | varchar(30)   | YES  |     | NULL    |       |
| BookName    | varchar(30)   | YES  |     | NULL    |       |
| AuthorName | varchar(30)   | YES  |     | NULL    |       |
| BookPrice  | int           | YES  |     | NULL    |       |
| Quantity   | int           | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

CUSTOMERORDERDETAILS TABLE:

Define the list of book and book details processes which are available in the bookstore.

Customer Order Details Table Structure:

```
mysql> describe CustomerOrderDetails;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| bookId     | int           | NO   |     | NULL    |       |
| Category   | varchar(30)   | YES  |     | NULL    |       |
| BookName    | varchar(30)   | YES  |     | NULL    |       |
| AuthorName  | varchar(30)   | YES  |     | NULL    |       |
| bookPrice   | int           | NO   |     | NULL    |       |
| quantity    | int           | YES  |     | NULL    |       |
| custId     | int           | YES  | MUL | NULL    |       |
| totalPrice  | int           | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

CUSTOMERSACCOUNT TABLE:

Define the customer's account details process which consists the fields accno, accpin, ifsccode, bankname, balance .

Customer Order Details Table Structure:

```
mysql> describe CustomersAccount;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Accno      | int           | NO   | PRI | NULL    |       |
| accpin     | int           | NO   |     | NULL    |       |
| ifsccode   | varchar(8)    | YES  |     | NULL    |       |
| Bankname   | varchar(30)   | NO   |     | NULL    |       |
| balance    | int           | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

ADMINACCOUNT TABLE:

Define the Admin's account details process where the customer's book price amount will be deposited into the admin's account. It consists the fields of accno and total price.

Admin Account Table Structure:

```
mysql> describe adminaccount;
+-----+-----+-----+-----+-----+-----+
| Field      | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| accno      | int  | NO   |     | NULL    |       |
| totalprice | int  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

DELIVERYDETAIL TABLE:

Define the customer's personal details for delivery which consists the fields of Custid, customerName, MobileNo, doorNo, StreetName, location, and Pincode.

Delivery Detail Table Structure:

```
mysql> describe deliverydetail;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| custid     | int       | NO   | PRI | NULL    |       |
| customerName | varchar(30) | NO   |     | NULL    |       |
| MobileNo   | varchar(10) | NO   |     | NULL    |       |
| doorNo     | varchar(20) | NO   |     | NULL    |       |
| StreetName | varchar(30) | NO   |     | NULL    |       |
| location   | varchar(30) | NO   |     | NULL    |       |
| pincode    | int       | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```


II.Java classfile overview:

This project consists of a 3 java class file...

1.ONLINEBOOKSTOREMAIN:

ADMIN AND USER MAIN METHOD CODE:

```
/******CORE JAVA PROJECT*****  
*****CSR CAPGEMINI TRAINING PROJECT*****  
*****EDUBRIDGE INDIA PRIVATE LIMITED*****  
*****PROJECT TITLE:ONLINE BOOK STORE*****  
*****@DONEBY S.DEEPIKA*****  
*****@UNDER THE GUIDANCE OF THE TRAINER-MRS.INDRAKKAMALLI**  
*/
```

* IN ONLINE BOOK STORE PROJECT

* MAIN OPERATIONS:

* 1.REGISTRATION -->CUSTOMER can perform that operation.

* 2.LOGIN -->Both ADMIN and CUSTOMER can perform that operation.

*/

*BookStoreMain class is having two major operations

*This will decides who is going to perform the operation

1.ADMIN

2.USER

case 1:

*If it is admin choice means it gives two choice for admin

*1.Admin Registration-->it allows to create new registration for admin

*2.Admin Login-->it allows to login the admin account

case 2:

*If it is user choice means it gives two choice for user

*1.User Registration-->it allows to registration for user

*2.User Login-->it allows to login the user account

```

*/

package com.onlinebookstore;

import java.util.Scanner;

//main class for Online Book Store

public class BookStoreMain {

    //getting input from the user

        static Scanner sc=new Scanner(System.in);
        public static void main(String[] args) {
            // TODO Auto-generated method stub
            while(true)
        {

System.out.println("Welcome To Online Book Store");
System.out.println("1.Admin");
System.out.println("2.User");
System.out.println("3.Exit");
System.out.println("enter your choice:");
int in=sc.nextInt();
switch(in)
{
case 1:
while(true)
{

System.out.println("1.Admin Login");
System.out.println("2.Exit");

System.out.println("enter your choice:");
int ch=sc.nextInt();
switch(ch)
{

case 1:System.out.println("login");
//perform the login operation
BookStoreProcess.adminLoginInfo();
break;

```

```

default:System.out.println("invalid input");
}
//for repetition purpose
System.out.println("do you want to continue the admin registration or login process yes/no");
//getting input
String input=sc.next();
if(input.equalsIgnoreCase("no"))//comparing input with the condition
{
break;
}
}
break;

```

```

case 2:
while(true)
{
System.out.println("1.User Registration");
System.out.println("2.User Login");

```

```

System.out.println("enter your choice:");
int ch=sc.nextInt();
switch(ch)
{
case 1:System.out.println("registration");
//perform the registration operation
BookStoreProcess.userRegistration();
break;
case 2:System.out.println("login");
//perform the login operation
BookStoreProcess.userLogin();
break;

```

```

default:System.out.println("invalid input");
}
//for repetition purpose
System.out.println("do you want to continue the user registration or login process yes/no");
//getting input
String input=sc.next();
if(input.equalsIgnoreCase("no"))//comparing input with the condition
{
break;
}
}

```

```

break;
case 3:
break;

default: System.out.println("invalid input");
//for repetition purpose
System.out.println("do you want to continue the admin or user process yes/no");
//getting input
String input=sc.next();
if(input.equalsIgnoreCase("no"))//comparing input with the condition
{
break;
}break;

}
}
}

```

DATABASE TABLE CREATION:

ADMIN REGISTRATION TABLE STRUCTURE:

```
mysql> describe admin_registration;
```

Field	Type	Null	Key	Default	Extra
emailid	varchar(30)	NO	PRI	NULL	
Username	varchar(30)	YES		NULL	
Password	varchar(30)	NO		NULL	
repass	varchar(20)	YES		NULL	

4 rows in set (0.02 sec)

USER REGISTRATION TABLE STRUCTURE:

```
mysql> describe user_registration;
```

Field	Type	Null	Key	Default	Extra
emailid	varchar(30)	NO	PRI	NULL	
Username	varchar(30)	YES		NULL	
Password	varchar(30)	NO		NULL	
repass	varchar(20)	YES		NULL	

4 rows in set (0.01 sec)

2.DBCONNECT.JAVA:

DATABASE CONNECTION CODE:

```
/**
 * Data base connection class-> it makes a connections between the spring tool
 * shoot and mysql
 *
 *      1.Driver->That implements the java databases connectivity (JDBC)
 * API
 *
 *      2.url->data base management system jdbc driver uses to connect to a
 * database
 *
 *      3.username-> user name of the database
 *
 *      4.password->pass word for the databases
 *
 *      5.forName()-> method is loading the driver dynamically loads a java class
 * at runtime
 *
 *      6.DriverManger->Is that class making connection to database by passing
 * arguments as a url ,username and password.
 */
package com.onlinebookstore;

import java.sql.Connection;

import java.sql.DriverManager;

public class DbConnect {

    //loading driver

    static String driver="com.mysql.cj.jdbc.Driver";

    //here the databases name is OnlineBookStore
```

```

//port number is 3306
static String url="jdbc:mysql://localhost:3306/OnlineBookStore";
//my sql username
static String username="root";
//mysql password
static String password="root";
static Connection con;
public static Connection getConnection()
{
try
{
//loading driver
Class.forName(driver);
//using this arguments building the connection between java and sql
con=DriverManager.getConnection(url, username, password);

}catch(Exception e)
{
e.printStackTrace();
}
return con; }}

```

3.BOOKSTOREPROCESS.JAVA

BOOK STORE PROCESS CODE:

ADMIN REGISTRATION:

FOR ADMIN:

1.The admin registration is done by backend process. Reason is only one admin for the whole application that why the admin registration is created by backed process.

- 2.This Admin Registration also contain the email id username and password.
- 3.There is no option is given to create the admin registration process at front end.

DATABASE OUTPUT:

Admin Registration Record is inserted in the Database:

```
mysql> SELECT * FROM admin_registration;
+-----+-----+-----+-----+
| emailid      | Username | Password | repass |
+-----+-----+-----+-----+
| deepi@gmail.com | deepi    | 123      | 123    |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

ADMIN LOGIN CODE:

//ADMIN LOGIN MODULE

```
/*
 * For this admin login process in the back end we had created one user name and password in the table called
 Admin Registration
 *
 * That user name and pass word can be used to login in the admin login module
 *
 * If the login process get successful means admin can perform all the operations
 *
 */
public static void adminLoginInfo()
{
    try
    {
        con=DbConnect.getconnection();
        //checking user name and pass word
        while(true)
        {
            System.out.println("Enter The UserName");
            username=sc.next();
            System.out.println("Enter The PassWord");
            password=sc.next();

            String checkusername="select* from admin_registration where username=?";
            pst=con.prepareStatement(checkusername);
```

```

pst.setString(1,username);
rst=pst.executeQuery();
if(rst.next())
{
String checkpass="select * from admin_registration where password =? ";
pst=con.prepareStatement(checkpass);
pst.setString(1,password);
rst=pst.executeQuery();
if(rst.next())
{
System.out.println("Login successfull");
System.out.println("1.viewing books");
System.out.println("2.adding books");
System.out.println("3.updating books");
System.out.println("4.Deleting books");
System.out.println("5.Viewing Customer order details");
System.out.println("6.viewing customers personal details");

System.out.println("Enter Your choice...");
int ch1=sc.nextInt();
switch(ch1)
{
case 1:System.out.println("You had the choosen book viewing process");
//perform the book display process
BookStoreProcess.viewBookInfo();
break;
case 2:System.out.println("you had choosen book adding Process");
//perform the book insert operation
BookStoreProcess.insertBookInfo();
break;
case 3:System.out.println("you had choosen updating process");
//perform the book update operation
BookStoreProcess.updateBookInfo();
break;
case 4:System.out.println("You had choosen book deleting Process");
//perform the book deleting operation
BookStoreProcess.deleteBookInfo();

case 5:System.out.println("You had choosen Customer order viewing process");
//performs the customers order display process
BookStoreProcess.viewCustomerOrderInfo();
break;
case 6:System.out.println("You had choosen Delivery Details of Customer process ");

```



```

//performs the customers personal info process
BookStoreProcess.viewCustomerDeliveryDetailsInfo();

break;
default: System.out.println("Invalid Choice");
break;
}break;
}
else
{
System.out.println("Invalid PassWord");
break;
}
}
else
{
System.out.println("Invalid User Name");

}
System.out.println("If You Are Already REGISTERED Then Go for once more chance");
System.out.println("Otherwise Go for REGISTRATION");
System.out.println("Do you want to go for ONE MORE CHANCE yes/no");
//ALREADY REGISTERED THEN GIVING ONE MORE CHANCE
String input=sc.next();
if(input.equalsIgnoreCase("no"))//comparing input with the condition
{
break;
}
}
}catch(Exception e)
{
e.printStackTrace();
}
}

```

ADMIN LOGIN JAVA OUTPUT:

```

Welcome To Online Book Store
1.Admin
2.User
3.Exit
enter your choice:
1
1.Admin Login
2.Exit
enter your choice:
1
login
Enter The UserName
DEEPI
Enter The PassWord
123
Login successfull

```

//ADMIN VIEW BOOK INFORMATION MODULE

```

private static void viewBookInfo() {
try
{
con=DbConnect.getconnection();
String sqldis="select *from BookStore";
pst=con.prepareStatement(sqldis);
rst=pst.executeQuery();
System.out.println("id\t\tcategory\t\tbookname\t\tauthorName\t\tprice\t\tquantity");
while(rst.next())
{
System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getString(3)+"\t"+rst.getString(4)+"\t"+rst.getString(5)+"\t"+rst.getString(6));
}

}catch(Exception e)
{
e.printStackTrace();
}
}

```

DATABASE OUTPUT:

```
mysql> select *from bookstore;
```

BookId	Category	BookName	AuthorName	BookPrice	Quantity
112	Suspense	The_Judges_House	Bram_Stocker	134	3
113	Poetry	Love_Poems	Ivo_Mosley	145	4
114	Story	Disney_Jungle_Book	Bernard_Cornwell	250	5
115	Crime	Death_On_The_Air	Ngaio_Marth	350	6
116	Dream	An_Into_To_Dream_Land	Bhagat_Singh	345	5
117	Awareness	A_Better_India	Narayana_Moorthy	450	7
118	Drama	A_Tale_Of_Two_Cities	Charles_Darwin	323	4
119	Comedy	The_Dancing_Partner	Jerome_K_Jerome	434	6
120	Science_Fiction	The_Sirens_of_Mars	Sara_Johnson	546	7

```
9 rows in set (0.01 sec)
```

```
//-----
```

```
//ADMIN INSERT BOOK INFORMATION MODULE
```

```
private static void insertBookInfo() {  
try  
{  
con=DbConnect.getconnection();  
System.out.println("Enter the BookName:");  
bookName=sc.next();  
String sql = "select *from BookStore where BookName=?";  
pst=con.prepareStatement(sql);  
pst.setString(1,bookName);  
rst=pst.executeQuery();  
if(!rst.next())  
{  
System.out.println("Enter the BookId");  
bookId=sc.nextInt();  
System.out.println("Enter the category");  
category=sc.next();  
System.out.println("Enter the Bookname");  
bookName=sc.next();  

```

```

System.out.println("Enter the AuthorName");
bookName=sc.next();
System.out.println("Enter the BookPrice");
bookPrice=sc.nextInt();
System.out.println("Enter the quantity");
quantity =sc.nextInt();
String sqlins="insert into BookStore values(?,?,?,?";
pst=con.prepareStatement(sqlins);
pst.setInt(1, bookId);
pst.setString(2,category);
pst.setString(3, bookName);
pst.setString(4, authorName);
pst.setInt(5,bookPrice);
pst.setInt(6, quantity);
int i=pst.executeUpdate();
if(i>0)
{
System.out.println("Inserted successfully");
}
}
else
{
System.out.println("enter the correct fields");
}
}
catch(Exception e)
{
e.printStackTrace();
}
}

//-----
-----

//ADMIN UPDATE BOOK INFORMATION MODULE

private static void updateBookInfo() {
try
{
con=DbConnect.getconnection();
System.out.println("Enter the BookId to update record:");
bookId=sc.nextInt();
String selbid="select *from BookStore where BookId=?";
pst=con.prepareStatement(selbid);

```

```

pst.setInt(1, bookId);
rst=pst.executeQuery();

if(rst.next())
{
while(true)
{

System.out.println("which field do you want to update");
System.out.println("1.Update only BookName");
System.out.println("2.Update only Bookprice");
System.out.println("3.Update only quantity");
System.out.println("4.Update Bookname,BookPrice,Quantity");
System.out.println("5.exit");
System.out.println("Enter the choices");
int choice=sc.nextInt();
switch(choice)
{
case 1:
System.out.println("Enter the Bookname to change");
bookName=sc.next();
String upd="update BookStore set bookname=? where bookid=?";
pst=con.prepareStatement(upd);
pst.setString(1,bookName);
pst.setInt(2,bookId);
int i=pst.executeUpdate();
if(i>0)
{
System.out.println("Record updated");
}
break;
case 2:
System.out.println("Enter the Bookprice to change");
bookPrice=sc.nextInt();
String updpri="update BookStore set bookprice=? where bookname=?";
pst=con.prepareStatement(updpri);
pst.setInt(1, bookPrice);
pst.setString(2, bookName);
int j=pst.executeUpdate();
if(j>0)
{
System.out.println("Record updated");
}
}
}
}

```

```

break;
case 3:
System.out.println("Enter the quantity to change");
quantity=sc.nextInt();
String updquan="update BookStore set quantity=? where bookname=?";
pst=con.prepareStatement(updquan);
pst.setInt(1, quantity);
pst.setString(2, bookName);
int k=pst.executeUpdate();
if(k>0)
{
System.out.println("Record updated");
}
break;
case 4:
System.out.println("Enter the Bookname to change");
bookName=sc.next();
System.out.println("Enter the Bookprice to change");
bookPrice=sc.nextInt();
System.out.println("Enter the Quantity to change");
quantity=sc.nextInt();
String updall="update BookStore set bookname=?,bookprice=?,quantity=? where bookid=?";
pst=con.prepareStatement(updall);
pst.setString(1, bookName);
pst.setInt(2, bookPrice);
pst.setInt(3, quantity);
pst.setInt(4, bookId);
int l=pst.executeUpdate();
if(l>0)
{
System.out.println("Record updated");
}
break;
case 5:
break;
default:System.out.println("Invalid input");
}
System.out.println("Dou you want to update again-yes/no");
String input=sc.next();
if(input.equalsIgnoreCase("no"))
{
break;
}
}}

```

```
else {  
    System.out.println(bookId+ "id not found");  
  
}  
}
```

```
catch(Exception e)  
{  
    e.printStackTrace();  
}  
}
```

```
//-----  
-----
```

```
//ADMIN DELETING BOOK INFORMATION MODULE
```

```
private static void deleteBookInfo() {  
    try  
    {  
        con=DbConnect.getConnection();  
        System.out.println("Enter the BookId:");  
        bookId=sc.nextInt();  
        String sql="select * from BookStore where bookid=?";  
        pst=con.prepareStatement(sql);  
        pst.setInt(1, bookId);  
        rst=pst.executeQuery();  
        if(rst.next())  
        {  
            String del="delete from BookStore where bookid=?";  
            pst=con.prepareStatement(del);  
            pst.setInt(1, bookId);  
            int i=pst.executeUpdate();  
            if(i>0)  
            {  
                System.out.println("Record is deleted");  
            }  
        }  
        else {  
            System.out.println("Record is not found");  
        }  
    }  
}
```

```

catch(Exception e)
{
    e.printStackTrace();
}
}

//-----
-----

//ADMIN VIEW CUSTOMER ORDER INFORMATION MODULE

private static void viewCustomerOrderInfo() {
// TODO Auto-generated method stub
try
{
    con=DbConnect.getConnection();
    String sqlorderdis="select *from CustomerOrderDetails";
    pst=con.prepareStatement(sqlorderdis);
    rst=pst.executeQuery();
    System.out.println("Id\t\tCategory\t\tBookName\t\tAuthorName\t\tbookPrice\t\tquantity\t\tcustId\t\ttotalprice");
    while(rst.next())
    {
        System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getString(3)+"\t"+rst.getString(4)
        +"\t"+rst.getInt(5)+"\t"+rst.getInt(6)+"\t"+rst.getInt(7)+"\t"+rst.getInt(8));
    }

}

}catch(Exception e)
{
    e.printStackTrace();
}

}
//_____
_____
_____

```



```
//ADMIN VIEW CUSTOMER PERSONAL INFORMATION MODULE
```

```
private static void viewCustomerDeliveryDetailsInfo() {  
// TODO Auto-generated method stub
```

```
// TODO Auto-generated method stub
```

```
try
```

```
{
```

```
con=DbConnect.getConnection();
```

```
String deliverydetail="select *from DeliveryDetails";
```

```
pst=con.prepareStatement(deliverydetail);
```

```
rst=pst.executeQuery();
```

```
System.out.println("Id\ttcustname\ttmobno\ttdoorno\ttstreetname\t\tlocation\t\tpincode  
");
```

```
while(rst.next())
```

```
{
```

```
System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getInt(3)+"\t"+rst.getInt(4)+"\t"+r  
st.getString(5)+"\t"+rst.getString(6)+"\t"+rst.getInt(7));
```

```
}
```

```
}catch(Exception e)
```

```
{
```

```
e.printStackTrace();
```

```
}
```

```
}
```

```
//
```

```
// USER REGISTRATION MODULE
```

```
/**
```

```
* User Registration
```

```
*
```

```
* Registration Modules-> User Registration() ->It takes informations from customer and saves in  
a database.
```

```
*
```

```

* ->first it checks the registration is already done by the same mail id or not.
* if it is not then ask further information regarding the registration...
*
* ->User name and password...while setting password confirmation also done.
*
*
*
* */

```

```

public static void userRegistration() {
try
{
con=DbConnect.getConnection();
System.out.println("enter your email id");
emailId=sc.next();
//fetching email id from user registration
String selemail="select * from user_registration where emailid=?";
pst=con.prepareStatement(selemail);
pst.setString(1, emailId);
rst=pst.executeQuery();
if(!rst.next())
{
System.out.println("Enter Your User Name");
username=sc.next();
System.out.println("Enter Your Pass Word");
password=sc.next();
while(true)
{
System.out.println("Re Enter the Pass Word");
String repass=sc.next();
//Checking both passwords are same or not
if(repass.equals(password))
{
String regsql="insert into user_registration values(?,?,?,?)";
pst=con.prepareStatement(regsql);
pst.setString(1, emailId);
pst.setString(2, username);
pst.setString(3, password);
pst.setString(4, repass);
int i=pst.executeUpdate();
if(i>0)
{
System.out.println("REGISTERED SUCCESSFULLUY...");
System.out.println("THANK YOU FOR YOUR REGISTRATION...");

```

```

break;
}}
else{
System.out.println("recheck the re entered pass word");
} //while loop closing
} //if statement closing
else
{
System.out.println("This Email id already exists try with different name");}
} catch (Exception e)
{
e.printStackTrace();

}
}

```

DATABASE OUTPUT:

```
mysql> select *from user_registration;
```

emailid	Username	Password	repass
bismi@gmail.com	bismi	bismi8999	bismi8999
deepisuji@gmail.com	deepi	dee123	dee123
jay@gmail.com	jay	jay123	jay123
kavi@gmail.com	kavi	kavi767	kavi767

4 rows in set (0.00 sec)

```
//-----
-----
```

//USER LOGIN MODULE

```

/*
* In this User login Module in the back end we had created one user name and password for
the user
*
* That user name and pass word can be used to login in the User login module
*
* If the login process get successful means user can perform all the operations
**/
public static void userLogin() {
try
{
con=DbConnect.getConnection();

```

```

//checking user name and pass word
while(true)
{
    System.out.println("Enter The UserName");
    username=sc.next();
    System.out.println("Enter The PassWord");
    password=sc.next();

    String checkusername="select* from user_registration where username=?";
    pst=con.prepareStatement(checkusername);
    pst.setString(1,username);
    rst=pst.executeQuery();
    if(rst.next())
    {
        String checkpass="select * from user_registration where password =? ";
        pst=con.prepareStatement(checkpass);
        pst.setString(1,password);
        rst=pst.executeQuery();
        if(rst.next())
        {
            System.out.println("LOGIN SUCCESSFULL");
            while(true)
            {

                System.out.println("1.viewbooks");
                System.out.println("2.BookCancellation");

                System.out.println("Enter Your choice...");
                int ch1=sc.nextInt();
                switch(ch1)
                {
                    case 1: System.out.println("You had choosen viewing the book Process");
                    //perform the book order operation
                    BookStoreProcess.BookView();
                    break;
                    case 2: System.out.println("You had choosen the book cancellation process");
                    //perform the book cancel operation
                    BookStoreProcess.BookCancellation();
                    break;
                    default: System.out.println("Invalid Choice");

                }
                System.out.println("Do you want to continue the userprocess-yes/no");
            }
        }
    }
}

```

```

String input=sc.next();
if(input.equalsIgnoreCase("no"))
{
break;
}

}

}

else
{
System.out.println("Invalid PassWord");
break;
}
}
else
{
System.out.println("Invalid User Name");

}
System.out.println("If You Are Already REGISTERED Then Go for once more chance");
System.out.println("Otherwise Go for REGISTRATION");
System.out.println("Do you want to go for ONE MORE CHANCE yes/no");
//ALREADY REGISTERED THEN GIVING ONE MORE CHANCE
String input=sc.next();
if(input.equalsIgnoreCase("no"))//comparing input with the condition
{
break;
}
}

}
catch(Exception e)
{
e.printStackTrace();
}
}

//-----
-----

//FOR USER PURPOSE.....BOOK VIEWING PROCESS MODULE IN THE BOOK STORE

```

```

/**
 * In this book viewing process user can see the whole books and book details which are
 * available in the Book Store
 *
 * when the user had seen the books if they decided to purchase then they can go for book
 * purchase module.
 */

public static void BookView() {

    try
    {
        con=DbConnect.getConnection();
        String sqldis="select *from BookStore";
        pst=con.prepareStatement(sqldis);
        rst=pst.executeQuery();
        System.out.println("id\t\tcategory\t\tbookname\t\tauthorName\t\tprice\t\tquantity");
        while(rst.next())
        {
            System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getString(3)+"\t"+rst.getString(4)+"\t"+rst.getString(5)+"\t"+rst.getString(6));
        }
        System.out.println("Do you want to order-yes/no");
        String input=sc.next();
        if(input.equalsIgnoreCase("yes"))
        {
            System.out.println("You had choosen purchasing the book Process");
            //perform the book canceling operation
            BookStoreProcess.BookPurchase();

        }

    }catch(Exception e)
    {
        e.printStackTrace();
    }
}

```

DATABASE OUTPUT:

```
mysql> select *from bookstore;
```

BookId	Category	BookName	AuthorName	BookPrice	Quantity
112	Suspense	The_Judges_House	Bram_Stocker	134	3
113	Poetry	Love_Poems	Ivo_Mosley	145	4
114	Story	Disney_Jungle_Book	Bernard_Cornwell	250	5
115	Crime	Death_On_The_Air	Ngaio_Marth	350	6
116	Dream	An_Into_To_Dream_Land	Bhagat_Singh	345	5
117	Awareness	A_Better_India	Narayana_Moorthy	450	7
118	Drama	A_Tale_Of_Two_Cities	Charles_Darwin	323	4
119	Comedy	The_Dancing_Partner	Jerome_K_Jerome	434	6
120	Science_Fiction	The_Sirens_of_Mars	Sara_Johnson	546	7

```
9 rows in set (0.01 sec)
```

```
//-----  
-----
```

```
//FOR USER PURPOSE-> BOOK PURCHASING MODULE
```

```
public static void BookPurchase() {  
  
    try {  
        while(true)  
        {  
            con=DbConnect.getconnection();  
            System.out.println("Enter the bookId:");  
            bookId=sc.nextInt();  
            String sql="select *from bookStore where bookId=?";  
            pst=con.prepareStatement(sql);  
            pst.setInt(1, bookId);  
            rst=pst.executeQuery();  
            if(rst.next())  
            {  
                System.out.println("Place your order");  
                //entering details for delivery  
                System.out.println("Enter your customerId:");  
                int custId=sc.nextInt();  
                System.out.println("Enter your customerName:");  
                String customerName=sc.next();  
                //System.out.println("Enter the BookQuantity:");  
                //quantity=sc.nextInt();  
                System.out.println("Enter Your MobileNo:");  
                String mobno=sc.next();  
                System.out.println("Enter Your full Address:");  
                System.out.println("Door No:");
```

```

String doorno=sc.next();
sc.nextLine();
System.out.println("Enter the Street Name:");
String streetname=sc.nextLine();
System.out.println("Enter Your Location:");
String location=sc.nextLine();
System.out.println("Enter the Pincode:");
int pincode=sc.nextInt();
String sqlper="insert into DeliveryDetail values(?,?,?,?,?,?,?)";
pst=con.prepareStatement(sqlper);
pst.setInt(1, custId);
pst.setString(2, customerName);
//pst.setInt(3, quantity);
pst.setString(3, mobno);
pst.setString(4, doorno);
pst.setString(5, streetname);
pst.setString(6, location);
pst.setInt(7, pincode);
int i=pst.executeUpdate();
if(i>0)
{System.out.println("your order information is taken:");

switch(bookId)
{
case 111:System.out.println("Your choosen this bookId" +bookId);

category="computer_programming";
bookName="Java_Complete_Reference";
authorName="Herbert_schildt";
bookPrice=123;
System.out.println("Enter the quantity");
quan=sc.nextInt();
quantity=quan;
totalprice=bookPrice*quan;
break;

case 112:System.out.println("Your choosen this bookid"+bookId);

category=" Suspense";
bookName="The_Judges_House";
authorName="Bram_Stocker";
bookPrice=134;

```



```
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 113: System.out.println("Your chosen this bookid"+bookId);
```

```
category="Poetry";  
bookName="Love_Poems";  
authorName="Ivo_Mosley";  
bookPrice=145;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 114: System.out.println("Your chosen this bookid"+bookId);
```

```
category="Story";  
bookName="Disney_Jungle_Book";  
authorName="Bernard_CornWell";  
bookPrice=250;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 115: System.out.println("Your chosen this bookid"+bookId);
```

```
category="Crime";  
bookName="Death_On_The_Air";  
authorName="Ngaio_Marth";  
bookPrice=350;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 116:System.out.println("Your choosen this bookid"+bookId);
```

```
category="Dream";  
bookName="An_Intro_To_DreamLand";  
authorName="Bhagat_Singh";  
bookPrice=345;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 117:System.out.println("Your choosen this bookid"+bookId);
```

```
category="Awareness";  
bookName="A_Better_India";  
authorName="Narayana_Murthy";  
bookPrice=450;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 118:System.out.println("Your choosen this bookid"+bookId);
```

```
category="Drama";  
bookName="A_Tale_of_Two_Cities";  
authorName="Charles_Darwin";  
bookPrice=323;  
System.out.println("Enter the quantity");  
quan=sc.nextInt();  
quantity=quan;  
totalprice=bookPrice*quan;  
break;
```

```
case 119:System.out.println("Your choosen this bookid"+bookId);
```

```
category="Comedy";
```

```

bookName="The_Dancing_Partner";
authorName="Jerome_K_Jerome";
bookPrice=434;
System.out.println("Enter the quantity");
quan=sc.nextInt();
quantity=quan;
totalprice=bookPrice*quan;
break;

```

```

case 120: System.out.println("Your chosen this bookid"+bookId);

```

```

category="Science_Fiction";
bookName="The_Sirens_Of_Mars";
authorName="Sara_Johnson";
bookPrice=546;
System.out.println("Enter the quantity");
quan=sc.nextInt();
quantity=quan;
totalprice=bookPrice*quan;
break;
default:
System.out.println("Invalid id");
}

```

```

String sqlins="insert into CustomerOrderDetails values(?,?,?,?,?,?,?)";
pst=con.prepareStatement(sqlins);
pst.setInt(1,bookId);
pst.setString(2,category);
pst.setString(3,bookName );
pst.setString(4,authorName);
pst.setInt(5,bookPrice);
pst.setInt(6,quantity);
pst.setInt(7,custId);
pst.setInt(8, totalprice);
int a=pst.executeUpdate();
if(a>0)
{
System.out.println("Select payment option");
System.out.println("1.Cash on Delivery");
System.out.println("2.Cash on Card");
int input=sc.nextInt();

switch(input){

```

```

case 1:System.out.println("YOU HAD CHOOSEN CASH ON DELELIVERY OPTION...");

System.out.println("YOUR ORDER WILL BE DISPATCH SOON....READY WITH YOU CASH
INHAND....THNAK YOU!");
break;

case 2:System.out.println("You had choosen cash on card payment option");
BookStoreProcess.Payment();
break;

}
break;
}

}
System.out.println("Do you want to order another one again-yes/no: ");
String input=sc.next();
if(input.equalsIgnoreCase("no"))
{
break;
}
break;
}
else
{
System.out.println("your choosen bookid is not there in the store...invalid option...so choose
the correct bookid");

}
break;
}}

catch(Exception e)
{
e.printStackTrace();
}
}

```

DATABASE OUTPUT:

```
mysql> select *from DELIVERYDETAILS;
```

custid	customerName	MobileNo	doorNo	StreetName	location	pincode
1001	DEEPI	9788810201	13	EAST STREET	TRICHY	621012
1003	KAVI	7347338778	23	KAVERY STREET	SALEM	728287

2 rows in set (0.00 sec)

```
mysql> SELECT *FROM CUSTOMERORDERDETAILS;
```

bookId	Category	BookName	AuthorName	bookPrice	quantity	custId	totalprice
112	Suspense	The_Judges_House	Bram_Stocker	134	2	1001	268
113	Poetry	Love_Poems	Ivo_Mosley	145	3	1003	435

2 rows in set (0.00 sec)

```
//-----  
-----
```

//FOR USER PURPOSE-> PAYMENT MODULE FOR AMOUNT PAY

```
private static void Payment() {
```

```
try {
```

```
con=DbConnect.getconnection();
```

```
System.out.println("go for payment");
```

```
System.out.println("enter your account number");
```

```
int accno=sc.nextInt();
```

```
System.out.println("enter your pin number");
```

```
int pinno=sc.nextInt();
```

```
String transacc="select accno from customersaccount where accno=?";
```

```
pst=con.prepareStatement(transacc);
```

```
pst.setInt(1, accno);
```

```
rst=pst.executeQuery();
```

```
if(rst.next())
```

```
{
```

```
String transpin="select accpin from customersaccount where accpin=?";
```

```
pst=con.prepareStatement(transpin);
```

```
pst.setInt(1, pinno);
```

```
rst=pst.executeQuery();
```

```
if(rst.next())
```

```

{
String balcheck="select balance from customersaccount where accno=? having balance>0";
pst=con.prepareStatement(balcheck);
pst.setInt(1,accno);
rst=pst.executeQuery();
if(rst.next())
{
String upd="update customersaccount set Balance=Balance-(select totalprice from
customerorderdetails where custId=?) where Accno=?";
pst=con.prepareStatement(upd);
pst.setInt(1,custId);
pst.setInt(2, accno);
int Z=pst.executeUpdate();
if(Z>0)
{
System.out.println("Your payment processed:");
String transadmin="insert into adminaccount values(?,?)";
pst=con.prepareStatement(transadmin);
pst.setInt(1,accno);
pst.setInt(2,totalprice);
int p=pst.executeUpdate();
if(p>0)
{
System.out.println("Your order is confirmed");
System.out.println("Do you want to See about your whole ORDER DETAILS ");
String o=sc.next();
if(o.equalsIgnoreCase("yes"))
{
System.out.println("Customerdetails");
BookStoreProcess.displayCustomerOrderDetails();
}

}

else
{
System.out.println("amount not inserted in adminaccount ");
}
}
else
{
System.out.println("Invalid pin number");
}
}

```

```

else
{
System.out.println("Insufficient balance");
}

}
else {
System.out.println("Account number not exists");
}
}}
catch(Exception e)
{
e.printStackTrace();
}}

```

DATABASE OUTPUT:

```

mysql> SELECT *FROM CUSTOMERSACCOUNT;
+-----+-----+-----+-----+-----+
| Accno  | accpin | ifsccode | Bankname  | balance |
+-----+-----+-----+-----+-----+
| 1808675 | 9504   | Cb234    | CanaraBank | 20000   |
| 9878675 | 2594   | Idbi234  | Idbi       | 50000   |
| 86767675 | 1234   | boi234   | BankOfIndia | 40000   |
| 86877675 | 1564   | bob234   | BankOfBaroda | 30000   |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

//FOR USER PURPOSE->DISPLAYING CUSTOMER ORDER DETAILS MODULE

```

public static void displayCustomerOrderDetails() {

try
{
con=DbConnect.getconnection();
System.out.println("Enter your custid:");
custId=sc.nextInt();
String display="select * from deliverydetail where custid=?";
pst=con.prepareStatement(display);
pst.setInt(1,custId);
rst=pst.executeQuery();

if(rst.next())

```

```

{
System.out.println("DeliveryDetail shown");
System.out.println("-----PERSONAL DETAILS-----");
System.out.println(" ");
System.out.println(" ");
System.out.println("custid\t\tcustomerName\tMobileNo\tDoorno\tStreetname\tLocation\tPin
code");
System.out.println(" ");
System.out.println(" ");
System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getString(3)+"\t"+rst.getString
(4)+"\t"+rst.getString(5)+"\t"+rst.getString(6)+"\t"+rst.getInt(7));
System.out.println(" ");
System.out.println(" ");
String orderch="select * from customerorderdetails where custid=?";
pst=con.prepareStatement(orderch);
pst.setInt(1, custId);
rst=pst.executeQuery();
while(rst.next())

{System.out.println("OrderDetail shown");
System.out.println(" ");
System.out.println("-----ORDER DETAILS-----");
System.out.println(" ");
System.out.println(" ");
System.out.println("bookId\t\tcategory\t\tbookname\t\tauthorname\t\tcustid");
System.out.println(" ");
System.out.println(" ");
System.out.println(rst.getInt(1)+"\t"+rst.getString(2)+"\t"+rst.getString(3)+"\t"+rst.getString
(4)+"\t"+rst.getInt(5));
System.out.println(" ");
System.out.println(" ");
String paymentch="select * from customerorderdetails where custid=?";
pst=con.prepareStatement(paymentch);
pst.setInt(1, custId);
rst=pst.executeQuery();
while(rst.next())

```



```

{
System.out.println(    );
System.out.println("-----PAYMENT DETAILS-----");
System.out.println("_____");
System.out.println("_____");
System.out.println("custId\t\tbookprice\tquantity\ttotalprice");
System.out.println("_____");
System.out.println("_____");

System.out.println(rst.getInt(7)+"\t\t"+rst.getString(5)+"\t\t"+rst.getInt(6)+"\t\t"+rst.getInt(8)+
"\t\t");
System.out.println("_____");
System.out.println("_____");
break;
}
}
}
else
{
System.out.println("You had given the wrong CustomerId....Please insert the correct
customerId...");
}
}
catch(Exception e)
{
e.printStackTrace();
}
}

```

JAVA OUTPUT :

*****Welcome To Online Book Store*****

1.ADMIN

2.USER

3.EXIT

enter your choice:

2

1.USER REGISTRATION

2.USER LOGIN

enter your choice:

2

LOGIN

Enter The UserName

DEEPI

Enter The Password

DEE123

LOGIN SUCCESSFULL

1.viewbooks

2.BookCancellation

Enter Your choice...

1

You had choosen viewing the book Process

id	category	bookname	authorName	price	quantity	
112	Suspense	The_Judges_House	Bram_Stocker	134		3
113	Poetry	Love_Poems	Ivo_Mosley	145	4	
114	Story	Disney_Jungle_Book	Bernard_Cornwell	250		5
115	Crime	Death_On_The_Air	Ngaio_Marth	350	6	
116	Dream	An_Into_To_Dream_Land	Bhagat_Singh	345	5	
117	Awareness	A_Better_India	Narayana_Moorthy	450		7
118	Drama	A_Tale_Of_Two_Cities	Charles_Darwin	323	4	
119	Comedy	The_Dancing_Partner	Jerome_K_Jerome	434	6	
120	Science_Fiction	The_Sirens_of_Mars	Sara_Johnson	546		7

Do you want to order-yes/no

YES

You had choosen purchasing the book Process

Enter the bookId:

113

Place your order

Enter your customerId:
1003
Enter your customerName:
KAVI
Enter Your MobileNo:
7347338778
Enter Your full Address:
Door No:
23
Enter the Street Name:
KAVERY STREET
Enter Your Location: |
SALEM
Enter the Pincode:
728287
your order information is taken:
Your choosen this bookid113
Enter the quantity
3
Select payment option
1.Cash on Delivery
2.Cash on Card
2
You had choosen cash on card payment option
go for payment
enter your account number
1808675
enter your pin number
9504
Your payment processed:
Your order is confirmed
Do you want to See about your whole ORDER DETAILS

YES
Customerdetails
Enter your custid:
1003
DeliveryDetail shown

DeliveryDetail shown

-----PERSONAL DETAILS-----						
custid	customerName	MobileNo	Doorno	Streetname	Location	Pincode
1003	KAVI	7347338778	23	KAVERY STREET	SALEM	728287

OrderDetail shown

-----ORDER DETAILS-----				
bookId	category	bookname	authorname	custid
113	Poetry	Love_Poems	Ivo_Mosley	145

-----PAYMENT DETAILS-----			
custId	bookprice	quantity	totalprice
1003	145	3	435

//-----

//FOR USER PURPOSE-> BOOK CANCELLATION MODULE

```
public static void BookCancellation()
{
    try
    {
```

```
        con=DbConnect.getConnection();
```

```
        System.out.println("Enter the bookId:");
```

```
        bookId=sc.nextInt();
```

```
        String sql="select *from customerorderdetails where bookId=?";
```

```
        pst=con.prepareStatement(sql);
```

```
        pst.setInt(1, bookId);
```

```
        rst=pst.executeQuery();
```

```
        if(rst.next())
```

```
        {
```

```
            System.out.println("enter your email id");
```

```
            emailId=sc.next();
```

```
            System.out.println("enter the user name");
```

```
            username=sc.next();
```

```
            System.out.println("enter the password");
```

```
            password=sc.next();
```

```
            System.out.println("enter the bookId");
```

```

bookId=sc.nextInt();
String chmail="select emailid from user_registration where emailid=? ";
pst=con.prepareStatement(chmail);
pst.setString(1,emailId);
rst=pst.executeQuery();
if(rst.next())
{
String chuser="select username from user_registration where username=? ";
pst=con.prepareStatement(chuser);
pst.setString(1,username);
rst=pst.executeQuery();
while(rst.next())
{
String chpass="select password from user_registration where password=? ";
pst=con.prepareStatement(chpass);
pst.setString(1,password);
rst=pst.executeQuery();
if(rst.next())
{
String canBook="delete from customerorderdetails where BookId=? ";
pst=con.prepareStatement(canBook);
pst.setInt(1,bookId);
int m=pst.executeUpdate();

if(m>0)
{
String candetail="delete from deliverydetail where custId=? ";
pst=con.prepareStatement(candetail);
pst.setInt(1,custId);
int n=pst.executeUpdate();
if(n>0)
{
System.out.println("Your cancellation is done");
break;
}}}
else
{
System.out.println("invalid email");
}

}
else
{
System.out.println("your not ordered any book...");
}

```

```
}
```

```
}
```

```
catch(Exception e) {  
e.printStackTrace();  
}}
```

