package com.simplilearn.capstone2.service;

import java.util.ArrayList;

import java.util.List;

import org.json.JSONObject;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.razorpay.Order;

import com.razorpay.RazorpayClient;

import com.razorpay.RazorpayException;

import com.simplilearn.capstone2.configuration.JwtRequestFilter;

import com.simplilearn.capstone2.dao.CartDao;

import com.simplilearn.capstone2.dao.MovieTicketDao;

import com.simplilearn.capstone2.dao.OrderDetailDao;

import com.simplilearn.capstone2.dao.UserDao;

import com.simplilearn.capstone2.entity.Cart;

import com.simplilearn.capstone2.entity.MovieTickets;

import com.simplilearn.capstone2.entity.OrderDetail;

import com.simplilearn.capstone2.entity.OrderInput;

import com.simplilearn.capstone2.entity.OrderTicketQuantity;

import com.simplilearn.capstone2.entity.TransactionDetails;

import com.simplilearn.capstone2.entity.User;

@Service

public class OrderDetailService {

private static final String ORDER\_PLACED = "Placed";

private static final String KEY = "rzp\_test\_IUZXQmB1qsPIjO";

private static final String KEY\_SECRET = "mw8lwZ73xTqmOyxBhSK1mhfK";

private static final String CURRENCY = "INR";

@Autowired

private OrderDetailDao orderDetailDao;

@Autowired

private MovieTicketDao mtDao;

@Autowired

private UserDao userDao;

@Autowired

private CartDao cartDao;

public void markOrderAsWatched(Integer orderId) {

OrderDetail orderDetail = orderDetailDao.findById(orderId).get();

if (orderDetail != null) {

orderDetail.setOrderStatus("Watched");

orderDetailDao.save(orderDetail);

}

}

public List<OrderDetail> getAllOrderDetails(String status) {

List<OrderDetail> orderDetails = new ArrayList<>();

if (status.equals("All")) {

orderDetailDao.findAll().forEach(x -> orderDetails.add(x));

} else {

orderDetailDao.findByOrderStatus(status).forEach(x -> orderDetails.add(x));

}

return orderDetails;

}

public List<OrderDetail> getOrderDetails() {

String currentUser = JwtRequestFilter.CURRENT\_USER;// will get the current username

User user = userDao.findById(currentUser).get(); // by this we will get all the userdetails

return orderDetailDao.findByUser(user);

// this method described inside orderDetailDao will return list of orders for

// that particular user

}

public void placeOrder(OrderInput orderInput, boolean isSingleTicketCheckout) {

List<OrderTicketQuantity> ticketQuantityList = orderInput.getOrderTicketQuantity();

for (OrderTicketQuantity o : ticketQuantityList) {

MovieTickets movieTicket = mtDao.findById(o.getTicketId()).get();// will return a product i.e movieTicket

// here

String currentUser = JwtRequestFilter.CURRENT\_USER;

User user = userDao.findById(currentUser).get();

OrderDetail orderDetail = new OrderDetail(orderInput.getFullName(), orderInput.getFullAddress(),

orderInput.getContactNumber(), orderInput.getAlternateContactNumber(), ORDER\_PLACED,

movieTicket.getDiscountedPrice() \* o.getQuantity(), movieTicket, user,

orderInput.getTransactionId());

// empty the cart after placing order

if (!isSingleTicketCheckout) {

List<Cart> cart = cartDao.findByUser(user);

cart.stream().forEach(x -> cartDao.deleteById(x.getCartId()));

}

orderDetailDao.save(orderDetail); // this irderdetail will get saved in database

}

}

// for payment gateway

public TransactionDetails createTransaction(Double amount) {

// Thing required for razorpay payment gateway:

// amount

// currency (usd/inr etc)

// key generated by razor pay portal

// secret key

//preparing JSON object to be passed insie .create() below--

JSONObject jsonObject = new JSONObject();

jsonObject.put("amount", amount\*100);

/\*as we are using INR the amount this method will recieve will be the smallest unit of currency-INR

i.e. paisa so to convert it into rupees we are placing \*100 here\*/

jsonObject.put("currency", CURRENCY);

// creating razorpay client

try {

RazorpayClient razorpayClient = new RazorpayClient(KEY, KEY\_SECRET);

Order order = razorpayClient.orders.create(jsonObject);

//.create(json object that contains amount & currency info)

TransactionDetails transactionDetails = prepareTransasctionDetails(order);

return transactionDetails;

} catch (RazorpayException e) {

e.printStackTrace();

System.out.println(e.getMessage());

}

return null;

}

public TransactionDetails prepareTransasctionDetails(Order order) {

String orderId = order.get("id");

String currency = order.get("currency");

Integer amount = order.get("amount");

TransactionDetails transactionDetails = new TransactionDetails(orderId, currency, amount, KEY);

return transactionDetails;

}

}