<?php

include './dbcon.php';

require './compute-perkm-rate.php';

$sellerID = mysqli\_real\_escape\_string($con, $\_POST['sID']);

$itemCategory = mysqli\_real\_escape\_string($con, $\_POST['itemCategory']);

$customerLat = mysqli\_real\_escape\_string($con, htmlspecialchars($\_POST['customerLat']));

$customerLng = mysqli\_real\_escape\_string($con, htmlspecialchars($\_POST['customerLng']));

$sql = "SELECT `computation\_method`, `flagdown\_rate`, `small\_item\_fee\_perkm`, `small\_item\_fee\_fixed`, `big\_item\_fee\_perkm`, `big\_item\_fee\_fixed`, `latitude`, `longitude` FROM `sellers\_company` WHERE `id` = '$sellerID'";

$result = $con->query($sql) or die($con->error);

$row = $result->fetch\_assoc();

extract($row);

$sqlFlagDown = "SELECT \* FROM `shipping\_flagdown` WHERE `id` = 1";

$runFlagDown = $con->query($sqlFlagDown) or die($con->error);

$rowFlagDown = $runFlagDown->fetch\_array();

$flagDownKm = doubleval($rowFlagDown['flagdown\_km']);

if ($computation\_method !== "Fixed" && $computation\_method !== "PerKM") {

$sql = "SELECT `amount` FROM `fees` WHERE `fee` = 'Shipping'";

$run\_sql = $con->query($sql) or die($con->error);

$row = $run\_sql->fetch\_assoc();

extract($row);

echo json\_encode(array("statusCode"=> 200, "fixedFee"=>doubleval($amount)));

}

else {

switch ($itemCategory) {

case "Small Item":

$origin = $customerLat . ',' . $customerLng;

$destination = $latitude . ',' . $longitude;

$totalDistance = computeDistance($origin, $destination);

if ($totalDistance <= $flagDownKm) {

echo json\_encode(array("statusCode"=> 201, "flagDownRate"=> $flagdown\_rate, "perKmFee"=> $small\_item\_fee\_perkm, "totalDistance" => $totalDistance, "shippingFee"=>$flagdown\_rate, "flagDownKM" => $flagDownKm));

}

else {

$multiplyDistance = doubleval($totalDistance) - doubleval($flagDownKm);

$shippingFee = doubleval($flagdown\_rate) + (doubleval($small\_item\_fee\_perkm) \* doubleval($multiplyDistance));

echo json\_encode(array("statusCode"=> 201, "flagDownRate"=> $flagdown\_rate, "perKmFee"=> $small\_item\_fee\_perkm, "totalDistance" => $totalDistance, "shippingFee"=>$shippingFee, "flagDownKM" => $flagDownKm));

}

break;

case "Large Item":

$origin = $customerLat . ',' . $customerLng;

$destination = $latitude . ',' . $longitude;

$totalDistance = computeDistance($origin, $destination);

if ($totalDistance <= $flagDownKm) {

echo json\_encode(array("statusCode"=> 202, "flagDownRate"=> $flagdown\_rate, "bigPerKm"=> $big\_item\_fee\_perkm, "totalDistance" => $totalDistance, "shippingFee"=>$flagdown\_rate, "flagDownKM" => $flagDownKm));

} else {

$multiplyDistance = doubleval($totalDistance) - doubleval($flagDownKm);

$shippingFee = doubleval($flagdown\_rate) + (doubleval($big\_item\_fee\_fixed) \* doubleval($multiplyDistance));

echo json\_encode(array("statusCode"=> 202, "flagDownRate"=> $flagdown\_rate, "bigPerKm"=> $big\_item\_fee\_perkm, "totalDistance" => $totalDistance, "shippingFee"=>$shippingFee, "flagDownKM" => $flagDownKm));

}

break;

default:

}

}

?>

JSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

$cart\_json = []

let creditLimit;

let creditTerms;

let agentCustomers;

var agentCustomersList = [];

let isForAgentCustomer,customerLat,customerLng;

async function loadCustomerDetails() {

try {

const response = await fetch('./includes/functions/load-customer-details.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

})

});

const data = await response.json();

if (data[0] == 200) {

$('.order-info\_\_name').text(data[1]);

$('.order-info\_\_number').text(data[2]);

$('.order-info\_\_address').text(data[3]);

$('.order-info\_\_addressLat').val(data[4]);

$('.order-info\_\_addressLong').val(data[5]);

} else {

Swal.fire({

title: 'Please login to continue.',

icon: 'warning',

confirmButtonColor: '#9d0001'

});

window.location.href = "login.html";

}

} catch (error) {

console.error(error);

}

}

async function checkRPAF(customerID) {

try {

const response = await $.ajax({

url: './includes/functions/check-rpaf-status.php',

data: customerID ? {customerID : customerID} : {},

method: "POST",

dataType: 'JSON'

});

const dataResult = JSON.parse(JSON.stringify(response));

if(dataResult.statusCode == 200) {

creditLimit = dataResult.creditLimit;

creditTerms = dataResult.creditTerms;

$('.js-btn-signup-rpaf').attr("disabled", true);

$('.rpaf-container').css("display", "none");

$('.total-rpaf-credit').text(formatter.format(creditLimit));

$('.total-rpaf-credit-days').text(creditTerms);

$('.js-btn-rpaf').css("display", "block");

$('.js-btn-split-payment').css("display", "block");

} else if(dataResult.statusCode == 201) {

$('.js-btn-rpaf').css("display", "none");

$('.js-btn-split-payment').css("display", "none");

$('.js-btn-signup-rpaf').attr("disabled", true);

} else if(dataResult.statusCode == 202) {

$('.js-btn-rpaf').css("display", "none");

$('.js-btn-split-payment').css("display", "none");

}

} catch(error) {

console.log(error);

// handle error

}

}

$( document ).ready(async function() {

// Customer Details

await loadCustomerDetails();

await checkRPAF();

const changeText = function (selectedText, selectedTextContainer) {

for (let i = 0; i < selectedText.length; i++) {

selectedText[i].addEventListener("click", function () {

selectedTextContainer.textContent = selectedText[i].textContent;

let dataID = $(this).attr('data-id');

$('#agentCustomerID').val(agentCustomers[dataID]['id'])

$('.order-info\_\_address').text(agentCustomers[dataID]['address'])

$('.order-info\_\_number').text(agentCustomers[dataID]['phone'])

$('.order-info\_\_name').text(agentCustomers[dataID]['name'])

$('#agentCustomerLat').val(agentCustomers[dataID]['latitude'])

$('#agentCustomerLng').val(agentCustomers[dataID]['longitude'])

$('#agentCustomerEmail').val(agentCustomers[dataID]['email'])

checkRPAF(agentCustomers[dataID]['id']);

isForAgentCustomer = 1;

renderCart()

});

}

};

// Check if agent and load its customer under agent

$.ajax({

url: './includes/functions/agent-customers.php',

success: function(data) {

let dataResponse = JSON.parse(data)

if(dataResponse.statusCode === 200) {

// Add to Customers List

agentCustomersList = dataResponse.dataList

const c = document.getElementById("customersDrop")

for(let key in agentCustomersList) {

let value = agentCustomersList[key]['name']

let dataID = agentCustomersList[key]['id']

const li = document.createElement("li")

li.textContent = value

li.classList.add("form\_\_dropdown-item");

li.classList.add("js-btn-item-credit-limit");

li.classList.add("dropdown-hover");

li.setAttribute('data-id', dataID)

c.appendChild(li)

const txt = document.querySelector(".js-text-area");

const btn = document.querySelectorAll(".js-btn-item-credit-limit");

changeText(btn, txt);

}

agentCustomers = dataResponse.dataList;

} else if(dataResponse.statusCode === 400) {

$('.order-info\_\_agent-customer').css('display', 'none')

$('.order-info\_\_agent-customer-address').css('display', 'none')

}

}

})

$(document).on('click', '.js-btn-search',function(){

Swal.fire({

title: 'Search Customer',

html: '<input type="text" class="form-search\_\_input js-input-popup-search"><ul class="form\_\_dropdown-list js-customer-drop" id="customerSearch"></ul>',

confirmButtonColor: '#9d0001',

focusConfirm: false,

showCancelButton: true,

}).then((result)=> {

if(result.isConfirmed) {

}

})

$.ajax({

url: './includes/functions/agent-customers.php',

success: function(data) {

let dataResponse = JSON.parse(data)

if(dataResponse.statusCode === 200) {

// Add to Customers List

agentCustomersList = dataResponse.dataList

const c = $("#customerSearch")

for(let key in agentCustomersList) {

console.log(agentCustomersList[key]['name'])

let value = agentCustomersList[key]['name']

let dataID = agentCustomersList[key]['id']

const li = document.createElement("li")

li.textContent = value

li.classList.add("form\_\_dropdown-item");

li.classList.add("js-btn-item-credit-limit");

li.classList.add("dropdown-hover");

li.setAttribute('data-id', dataID)

c.append(li)

const txt = document.querySelector(".js-text-area");

const btn = document.querySelectorAll(".js-btn-item-credit-limit");

changeText(btn, txt);

}

agentCustomers = dataResponse.dataList;

console.log(agentCustomers)

} else if(dataResponse.statusCode === 400) {

$('.order-info\_\_agent-customer').css('display', 'none')

$('.order-info\_\_agent-customer-address').css('display', 'none')

}

}

})

$(".js-input-popup-search").keyup(function(){

var searchkey= $(this).val();

var filterkey = searchkey.toUpperCase();

var li = $("#customerSearch li");

for (i = 0; i < li.length; i++) {

var txtValue = li[i].innerText;

if (txtValue.toUpperCase().indexOf(filterkey) > -1) {

li[i].style.display = "";

} else {

li[i].style.display = "none";

}

}

});

$(document).on('click', '.js-btn-item-credit-limit',function(){

Swal.close()

});

});

// headers

loadModule(".red-bar", "includes/pages/header-red-bar.php", "Header Redbar" , 1, '',0)

loadModule("#head\_2nd\_layer", "includes/pages/head-second.php", "Header 2nd" , 1, '',0)

loadModule("#head\_nav", "includes/pages/head-nav.php", "Header nav" , 1, '',0)

loadModule("#borger", "includes/pages/hamburger.php", "Burger" , 1)

// main

// var url\_string = window.location.href;

// var url = new URL(url\_string);

// var c = url.searchParams.get("seller");

// console.log(c);

// loadModule(".profile", "includes/pages/shop-on-seller-profile.php", "profile" , 1, c,0)

// loadModule(".filters", "includes/pages/shop-filters.php", "filters" , 1, '',1)

// loadModule(".products", "includes/pages/shop-products.php", "products" , 1, c,0)

// In your Javascript (external .js resource or <script> tag)

// footer

loadModule("#footer\_cont", "includes/pages/footer.php", "Footer" , 1, '',0)

$cart\_json= ( sessionStorage.getItem("\_session\_cart") != null ? JSON.parse( sessionStorage.getItem("\_session\_cart")) : [] ) ;

$address\_json= ( sessionStorage.getItem("\_session\_address") != null ? JSON.parse( sessionStorage.getItem("\_session\_address")) : [] ) ;

$remarks= ( sessionStorage.getItem("\_session\_remarks") != null ? ( sessionStorage.getItem("\_session\_remarks")) : '' ) ;

$('.cart-number').text($cart\_json.length)

renderCart()

renderTotal()

updateAddress()

$('#pick-up').prop('checked', true);

$('.pmethod\_card').hide()

$('.pmethod\_bank').hide()

$('.pmethod\_split').hide()

})

$('.js-btn-payment-method').on('click', function() {

$('.js-btn-payment-method').removeClass('order-info\_\_method--active')

$(this).addClass('order-info\_\_method--active')

var pmethod = $('.order-info\_\_block').find('.order-info\_\_method--active').data('pmethod')

$('.pmethod\_card').hide()

$('.pmethod\_bank').hide()

$('.pmethod\_split').hide()

if(pmethod == 'Bank Transfer'){

$('.pmethod\_bank').show()

$('.order-summary\_\_credit-limit').css("display", "none");

$('.order-summary\_\_credit-limit-days').css("display", "none");

}else if(pmethod == 'RPAF'){

let cartTotal = $('.summary-total').text().replace(/₱|,/g, '');

let rpafCredit = $('.total-rpaf-credit').text().replace(/₱|,/g, '');

let res = rpafCredit - cartTotal;

if(res < 0) {

Swal.fire(

{title: 'Your credit limit is not sufficient.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

// alert("Your credit limit is not sufficient.");

$('.js-btn-payment-method').removeClass('order-info\_\_method--active')

} else {

$('.order-summary\_\_credit-limit').css("display", "flex");

$('.order-summary\_\_credit-limit-days').css("display", "flex");

$('.pmethod\_card').show()

}

} else if(pmethod === "Split") {

$('.pmethod\_split').show()

let rpafBalance = $('.total-rpaf-credit').text();

$('.rpaf-balance').text(rpafBalance)

}

})

$('.js-amount-rpaf-input').on('keyup', function() {

let amountInput = parseFloat($(this).val())

let cartTotal = parseFloat($('.summary-total').text().replace(/₱|,/g, ''))

let rpafBalance = parseFloat($('.rpaf-balance').text().replace(/₱|,/g, ''))

let amountPaidinBank = cartTotal - amountInput

if(isNaN(amountPaidinBank)) {

$('.bank-remaining-balance').text(formatter.format(0))

} else {

$('.bank-remaining-balance').text(formatter.format(amountPaidinBank))

}

if(amountInput > rpafBalance) {

Swal.fire(

{title: 'Your credit limit is not sufficient.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

$(this).val("")

} else if(amountInput > cartTotal) {

Swal.fire(

{title: 'Amount cannot exceed total cart items.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

$(this).val("")

}

})

$('.js-btn-edit-address-save').on('click', function(){

// var name = $('#form\_name').val()

// var form\_email = $('#form\_email').val()

// var contact = $('#form\_contact').val()

// var city = $('#form\_city').val()

// var postal = $('#form\_postal').val()

// var address = $('#form\_address').val()

// var json\_arr = {

// name: name,

// email: form\_email,

// contact: contact,

// city: city,

// name: name,

// address: address

// }

// sessionStorage.setItem("\_session\_address", JSON.stringify( json\_arr) );

// location.reload();

})

$('.js-btn-standard-plus').on('click', function() {

renderTotal()

})

$('.js-btn-standard-minus').on('click', function() {

renderTotal()

})

$('.shipping-method\_\_input-count').on('change', function() {

renderTotal()

})

$('.form')

$('.btn-proceed-to-payment').on('click', function (){

var pmethod = $('.order-info\_\_block').find('.order-info\_\_method--active').data('pmethod')

console.log(pmethod);

if(pmethod == "Bank Transfer" && document.getElementById("imgFile").files.length == 0) {

// alert("Please upload bank proof.");

Swal.fire(

{title: 'Please upload bank proof.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

} else if(pmethod === "Split" && document.getElementById("imgFile2").files.length === 0) {

Swal.fire(

{title: 'Please upload bank proof.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

}

else if(pmethod === undefined) {

Swal.fire(

{title: 'Please select a payment method.',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

}

else {

$(document).on('scroll', function() {

window.scrollTo(0, 0);

});

$('.preloader').css("display", "block");

var id = $('[name="shipping-method"]:checked').attr('id')

var pmethod = $('.order-info\_\_block').find('.order-info\_\_method--active').data('pmethod')

if(id =='standard'){

var shipping = parseInt($(`#${id}`).data('shiping'))

var ship ='Standard Shipping'

var ship\_date = $('#shipping-date').val()

if($('#unloading').is(':checked' ) ){

var helper = parseInt($('.shipping-method\_\_input-count').val())

var helper\_fee = parseInt($('.shipping-method\_\_input-count').data('helper-fee'))

helper = helper \* helper\_fee

}

}else{

var shipping = 0

var ship ='Pick-up'

var helper = 0

var ship\_date = $('#shipping-date-pickup').val()

}

var voucherApplied = $('.voucher-id').val()

let voucherVal = $('#voucher-val').val()

if(voucherApplied == "") {

voucherApplied = 0;

voucherVal = 0;

}

$('.btn-proceed-to-payment').prop("disabled", true);

var formData = new FormData();

formData.append('info', JSON.stringify($address\_json));

formData.append('cart', JSON.stringify($cart\_json));

formData.append('ship\_fee', shipping);

formData.append('ship\_type', ship);

formData.append('ship\_date', ship\_date);

formData.append('pmethod', pmethod);

formData.append('voucherApplied', voucherApplied);

formData.append('voucherVal', voucherVal);

formData.append('remarks', $remarks);

formData.append('picture', $('#imgFile')[0].files[0]);

// formData.append('picture', $('input[type=file]')[0].files[0]);

if(pmethod === "Split") {

formData.append('rpafAmount', $('.js-amount-rpaf-input').val())

formData.append('bankAmount', $('.bank-remaining-balance').text().replace(/₱|,/g, ''))

formData.append('picture', $('#imgFile2')[0].files[0]);

}

// Agent Customer

if (isForAgentCustomer === 1) {

let agentCustomerID = $('#agentCustomerID').val()

let agentCustomerEmail = $('#agentCustomerEmail').val()

let agentCustomerName = $('.order-info\_\_name').text()

let agentCustomerPhone = $('.order-info\_\_number').text()

let agentCustomerAddress = $('.order-info\_\_address').text()

let agentCustomerLat = $('#agentCustomerLat').val()

let agentCustomerLng = $('#agentCustomerLng').val()

const agentKeys = ['ID', 'Email', 'Name', 'Phone', 'Address', 'Lat', 'Lng'];

for (let key of agentKeys) {

formData.append(`agentCustomer${key}`, eval(`agentCustomer${key}`));

}

}

$.ajax({

url: 'includes/functions/insert\_orders.php',

data: formData,

type: 'POST',

contentType: false,

processData: false,

success: function(data) {

// console.log(data)

data = JSON.parse(data)

if(data.status == 1){

$('.preloader').css("display", "none");

$(document).off('scroll');

sessionStorage.clear(); //reset session cart

$('.js-popup-success').removeClass('u-hidden')

$('.split-method').addClass('u-hidden')

$('.result-name').text(data.name)

$('.result-order-no').text(data.ref )

if(data.pmethod == "RPAF") {

$('.js-btn-receipt-form').attr("value",data.ref)

$('.js-btn-receipt-form').css("display","block")

$('.result-image-proof').css("display","none")

}

if(data.pmethod === "Split") {

$('.split-method').removeClass('u-hidden')

$('.result-total-rpaf').text(formatter.format(data.rpafAmount))

$('.result-total-bank').text(formatter.format(data.bankAmount))

}

$('.result-track-no').text(data.ref )

$('.result-email').text(data.email)

$('.result-address').text(data.address)

$('.result-city').text($address\_json['city'])

$('.result-contact').text(data.phone)

$('.result-pmethod').text(data.pmethod)

$('.result-image-proof').attr("src","uploads/" + data.fileName)

$('.result-total-cost').text(formatter.format(data.total\_fee ))

}else{

$('.preloader').css("display", "none");

$(document).off('scroll');

Swal.fire(

{title: 'Something went wrong, order Failed!',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

// alert('Something went wrong, order Failed!')

}

},

error: function(request, error) {

$('.preloader').css("display", "none");

$(document).off('scroll');

console.log(request)

console.log(error)

Swal.fire(

{title: 'Something went wrong',

icon: 'error',

confirmButtonColor: '#9d0001'}

)

// alert("Something went wrong");

}

});

}

})

$('.js-btn-popup-success-close').on('click', function (){

document.location ='index.html'

})

// Functions

function updateAddress(){

console.log($address\_json['name'])

// $address\_json.forEach( (item) => {

$('.order-info\_\_name').text($address\_json['name'])

$('.order-info\_\_number').text($address\_json['contact'])

$('.order-info\_\_address').text($address\_json['address'])

// })

}

function renderTotal(){

setTimeout(function() {

var id = $('[name="shipping-method"]:checked').attr('id')

// console.log(id)

if(id =='standard'){

var shipping = parseFloat($(`#${id}`).attr('data-shiping'))

}else{

var shipping = 0

}

if( id =='standard' ){

var helper = 0

// var helper = parseInt($('.shipping-method\_\_input-count').val())

// var helper\_fee = parseInt($('.shipping-method\_\_input-count').data('helper-fee'))

// helper = helper \* helper\_fee

}else{

var helper = 0

}

var checkVoucher = $('.summary-voucher').text();

if(checkVoucher == "₱0.00" || checkVoucher == "-") {

var voucher = 0;

} else {

var voucher = parseFloat($('#voucher-val').val());

}

// standard

// pick-up

let cartTotal = 0;

$cart\_json.forEach( (item, index) =>{

cartTotal += parseFloat(item.price \* item.qty );

})

$('.summary-subtotal').text(formatter.format(cartTotal))

$('.summary-voucher').text(formatter.format(voucher))

// $('.summary-helper').text(formatter.format(helper))

$('.summary-shipping').text(formatter.format(shipping))

$('.summary-total').text(formatter.format((cartTotal + shipping) - voucher))

$('.cart\_\_total-price').text(formatter.format(cartTotal))

}, 200);

}

function renderCart() {

let cartString = "";

let cartTotal = 0;

var html = "";

var totalQty = 0;

var load = 0;

let cartItems = $cart\_json.length;

if(cartItems < 1) {

Swal.fire({

title: 'No cart items found',

icon: 'warning',

confirmButtonColor: '#9d0001'

})

.then((result) => {

if(result.isConfirmed) {

window.location.href = "index.html";

}

})

return false;

}

let seller\_id = $cart\_json[0].store;

let itemCategory = $cart\_json[0].sizeCategory;

$cart\_json.forEach( (item,index) => {

html += `

<div class="order-summary\_\_product">

<svg

onclick="removeCart(${item.id})"

class="order-summary\_\_btn-remove js-btn-order-summary-remove"

>

<use xlink:href="img/sprite.svg#icon-x"></use>

</svg>

<div class="order-summary\_\_img-container">

<img

src="img/products/${item.productImage}"

alt="Product 1 image"

class="order-summary\_\_img"

/>

</div>

<h4 class="order-summary\_\_heading heading-4">

${item.productName}

<span>( ${item.qty}x)</span>

</h4>

<button type="button" class="btn-checkout-change-qty js-change-qty" value="${item.id}">Change qty</button>

<p class="order-summary\_\_price"> ${formatter.format(parseFloat(item.qty \* item.price))}</p>

</div>

`

cartTotal += parseFloat(item.price \* item.qty );

totalQty += parseInt(item.qty);

});

// Quantity Multiply Shipping

load = totalQty / 1200;

let newShipping = 0;

if(load > 0) {

var id = $('[name="shipping-method"]:checked').attr('id')

// if(id =='standard'){

customerLat = $('#lat').val()

customerLng = $('#long').val()

if(isForAgentCustomer === 1) {

customerLat = $('#agentCustomerLat').val()

customerLng = $('#agentCustomerLng').val()

}

let fd = new FormData();

fd.append('sID', seller\_id)

fd.append('customerLat', customerLat)

fd.append('customerLng', customerLng)

fd.append('itemCategory', itemCategory)

// Dynamic Shipping Fee

$.ajax({

url: './includes/functions/get-shipping-fee-v2.php',

dataType: 'JSON',

method: 'POST',

data: fd,

processData: false,

contentType: false,

success: function(data) {

let dataResponse = data;

if(dataResponse.statusCode === 201) {

let itemPerKM = dataResponse.perKmFee;

let totald = dataResponse.totalDistance;

let qtyShipping = itemPerKM \* totald

console.log(qtyShipping)

$(`#standard`).attr('data-shiping', qtyShipping);

$(`#${id}`).attr('data-shiping', qtyShipping);

document.getElementById('shipfee').innerHTML = `${formatter.format(qtyShipping)}`;

}

else if(dataResponse.statusCode === 202){

let itemPerKM = dataResponse.bigPerKm;

let totald = dataResponse.totalDistance;

let qtyShipping = itemPerKM \* totald

console.log(qtyShipping)

$(`#standard`).attr('data-shiping', qtyShipping);

$(`#${id}`).attr('data-shiping', qtyShipping);

document.getElementById('shipfee').innerHTML = `${formatter.format(qtyShipping)}`;

}

else if(dataResponse.statusCode === 200) {

let shipRateFixed = dataResponse.fixedFee;

$(`#standard`).attr('data-shiping', shipRateFixed);

shipping\_fee = parseFloat(shipRateFixed);

let round = Math.ceil(load);

let dynamicSFee = shipping\_fee;

newShipping = dynamicSFee \* round;

$(`#${id}`).attr('data-shiping', newShipping);

document.getElementById('shipfee').innerHTML = `${formatter.format(newShipping)}`;

}

}

})

// }

}

html += `

<div class="order-summary\_\_calculations">

<div class="order-summary\_\_calculations-block">

<div class="order-summary\_\_calculations-price">

<p>Subtotal</p>

<p class="summary-subtotal">-${formatter.format(cartTotal)}</p>

</div>

<div class="order-summary\_\_calculations-price">

<p>Shipping</p>

<p class="summary-shipping">${formatter.format(newShipping)}</p>

</div>

<div class="order-summary\_\_calculations-price">

<p>Discount/Voucher</p>

<p class="summary-voucher">-</p>

<input type='hidden' id='voucher-val' value=''>

</div>

</div>

<div class="order-summary\_\_calculations-total">

<p>Total</p>

<p class="summary-total">${formatter.format(cartTotal)}</p>

</div>

<div class="order-summary\_\_credit-limit">

<p>BuildCredit Limit</p>

<p class="total-rpaf-credit">${formatter.format(creditLimit)}</p>

</div>

<div class="order-summary\_\_credit-limit-days">

<p>BuildCredit Date Limit</p>

<p class="total-rpaf-credit-days">${creditTerms}</p>

</div>

</div>

`;

$('.order-summary').html(html)

renderTotal()

}

$(document).on('change', '#shipping-date', function(){

var pDate = document.getElementById('shipping-date').value;

var today = new Date();

let threeDays = new Date(today);

let pickedDate = new Date(pDate);

threeDays.setDate(today.getDate()+4);

if(pickedDate.getTime() < threeDays.getTime()) {

const date = new Date();

const newDate = date.getDate() + 4;

date.setDate(newDate);

const inputDateShippingStandard = document.querySelector(

".js-input-date-shipping-standard"

);

if (inputDateShippingStandard != undefined) {

inputDateShippingStandard.valueAsDate = date;

}

Swal.fire(

{title: 'You can only choose four days from today.',

icon: 'info',

confirmButtonColor: '#9d0001'}

)

// alert("You can only choose three days from today.");

// var setDate = threeDays.getFullYear() + '/' + (threeDays.getMonth() + 1) + '/' + threeDays.getDate();

}

})

$(document).on('change', '#shipping-date-pickup', function(){

var pDate = document.getElementById('shipping-date-pickup').value;

var today = new Date();

let threeDays = new Date(today);

let pickedDate = new Date(pDate);

// threeDays.setDate(today.getDate()+4);

threeDays.setDate(today.getDate());

if(pickedDate.getTime() < threeDays.getTime()) {

const date = new Date();

// const newDate = date.getDate() + 4;

const newDate = date.getDate();

date.setDate(newDate);

const inputDateShippingPickup = document.querySelector(

".js-input-date-shipping-pickup"

);

if (inputDateShippingPickup != undefined) {

inputDateShippingPickup.valueAsDate = date;

}

Swal.fire(

// {title: 'You can only choose four days from today.',

{title: 'Invalid date.',

icon: 'info',

confirmButtonColor: '#9d0001'}

)

// alert("You can only choose three days from today.");

// var setDate = threeDays.getFullYear() + '/' + (threeDays.getMonth() + 1) + '/' + threeDays.getDate();

}

})

function removeCart(id){

// console.log(id)

// console.log($cart\_json)

if($cart\_json.find(el => el.id == id )){

var removeCartIndex = $cart\_json.find(el => el.id == id )

// console.log(removeCartIndex)

var spliced = $cart\_json.splice(removeCartIndex, 1);

// console.log(spliced)

// console.log($cart\_json)

sessionStorage.setItem("\_session\_cart", JSON.stringify( $cart\_json) );

renderCart()

}else{

console.log('not found')

}

}

$(document).on('click', '.js-change-qty', function() {

let itemID = this.value

Swal.fire({

title: 'Change Quantity',

icon: 'info',

html: '<input type="number" class="form-search\_\_input js-input-popup-custom">',

confirmButtonColor: '#9d0001',

focusConfirm: false,

showCancelButton: true,

}).then((result)=> {

if(result.isConfirmed) {

let customQty = parseInt($('.js-input-popup-custom').val());

const maxQty = 1200;

const minQty = 1;

if($cart\_json.find(el => el.id == itemID)) {

let index = $cart\_json.findIndex(el => el.id == itemID )

let quantityLimit = parseInt($cart\_json[index].qtyLimit)

if(customQty > quantityLimit) {

Swal.fire(

{title: 'Order quantity limit reached',

html: 'Quantity Limit: ' + quantityLimit,

icon: 'error',

confirmButtonColor: '#9d0001'}

)

return false;

}

if(customQty > maxQty){

customQty = maxQty

}

if(customQty < minQty){

customQty = 1

}

$cart\_json[index].qty = customQty

sessionStorage.setItem("\_session\_cart", JSON.stringify( $cart\_json) );

renderCart()

}

}

})

})

function loadModule(element, link, name, display, param, triggers){

$( element ).load( link+"?data="+param, function() {

if(display ==1)

console.log( name+" was displayed" );

else if(display ==2)

alert( name+" was displayed" );

if(triggers ==1){

$(document).ready(function() {

$('.js-example-basic-single').select2();

});

}

});

}

$(document).on('click', '.js-btn-shop-voucher', function(){

var shipVoucher = $('.js-btn-input-voucher').val();

$.ajax({

url: './includes/functions/verify-voucher.php',

method: 'POST',

dataType: 'JSON',

data: {

shipVoucher:shipVoucher

},

success:function(data) {

data = JSON.parse(JSON.stringify(data));

if(data.statusCode == 200) {

const { seller\_id, vType, amount, voucherID } = data;

// Check Seller ID through cart

let vSeller = seller\_id;

if(vSeller == "" || $cart\_json.find(el => el.store == vSeller)) {

console.log(vSeller);

if(vType == "percent") {

var subTotal = $('.summary-subtotal').text().replace(/₱|,/g, '');

var vPercent = amount;

var vTotal = subTotal \* vPercent

$('.summary-voucher').text(vTotal);

$('#voucher-val').val(vTotal);

} else {

$('.summary-voucher').text(amount);

$('#voucher-val').val(amount);

}

$('.js-btn-input-voucher').attr("disabled", "true");

$('.js-btn-shop-voucher').attr("disabled", "true");

$('.js-btn-shop-voucher').css("color", "gray");

$('.voucher-id').val(voucherID);

renderTotal();

Swal.fire(

{title: 'Voucher Applied.',

icon: 'success',

confirmButtonColor: '#9d0001'}

)

}

else if($cart\_json.find(el => el.store != vSeller)) {

Swal.fire({

title: 'Voucher cannot be applied for this seller',

icon: 'error',

confirmButtonColor: '#9d0001'

})

}

}

else if(data.statusCode == 201) {

Swal.fire(

{title: 'Voucher does not exists.',

icon: 'info',

confirmButtonColor: '#9d0001'}

)

// alert("Voucher does not exists.");

}

}

})

})

// Create our number formatter.

const formatter = new Intl.NumberFormat('en-US', {

style: 'currency',

currency: 'PHP',

// These options are needed to round to whole numbers if that's what you want.

//minimumFractionDigits: 0, // (this suffices for whole numbers, but will print 2500.10 as $2,500.1)

//maximumFractionDigits: 0, // (causes 2500.99 to be printed as $2,501)

});

$('.form-customer-address-btn').on('click', function(){

var lat = $('#setLat').val();

var long = $('#setLong').val();

var placeName = $('#searchLocation').val();

var agentCustomerID = $('#agentCustomerID').val();

// renderTotal();

$.ajax({

url: './includes/functions/update-customer-address.php',

data: {

lat: lat,

long: long,

placeName: placeName,

agentCustomerID: agentCustomerID

},

async: true,

method: 'POST',

success: function(data) {

var data = JSON.parse(data);

if (data.statusCode == 200) {

Swal.fire({

title: 'Location updated successfully.',

icon: 'success',

confirmButtonColor: '#9d0001'

}).then((result) => {

if (result.isConfirmed) {

// $('.order-info\_\_address').val(placeName);

$('.order-info\_\_address').text(placeName);

$('#agentCustomerLat').val(lat);

$('#agentCustomerLng').val(long);

$('.js-popup-edit-address').addClass('u-hidden');

renderCart();

// location.reload();

}

})

} else if (data.statusCode == 201) {

Swal.fire({

title: 'Something went wrong. Please try again.',

icon: 'error',

confirmButtonColor: '#9d0001'

})

}

}

})

})