const parentElement = document.getElementById("customer");

const tableBody = document.getElementById("tableBody");

$(document).ready(function () {

// headers

loadModule(

".red-bar",

"includes/pages/header-red-bar.php",

"Header Redbar",

1

);

loadModule(

"#head\_2nd\_layer",

"includes/pages/head-second.php",

"Header 2nd",

1

);

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

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

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

getCustomers();

$(document).on("change", "#customer", async function () {

const customerId = this.value;

try {

const response = await fetch(

"./includes/functions/logistics-payment/getCustomerData.php",

{

method: "POST",

body: new URLSearchParams({

customerId: customerId,

}),

}

);

const data = await response.json();

// Create DataTable after populating the customers dropdown

let table = new DataTable("#datatable", {

responsive: true,

data: data,

bDestroy: true,

columns: [

{ data: "orderNo" },

{ data: "customerName" },

{

data: "logisticFeeRate",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

return `<span class="table-invoice-amount">${data}</span>`;

},

},

{

data: "totalQty",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

return `<span class="table-invoice-amount">${data}</span>`;

},

},

{

data: "grandTotal",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

return `<span class="table-invoice-amount">${data}</span>`;

},

},

{

data: "balance",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

return `<span id="remainingBalance" class="table-remaining-balance">${data}</span>

`;

},

},

{

data: "deliveredPicture",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

// return `<img src="https://buildmart.ph/img/delivery/${data}" class="delivered-picture">`;

return `<span class="view-dr" data-value="${row.orderNo}">View</span>`;

},

},

{

data: "dateDelivered",

render: function (data, type, row) {

// Add attributes or classes to the cell's data

return `<span class="table-invoice-amount">${data}</span>`;

},

},

{

data: "orderNo",

render: function (data, type, row) {

// Check if the balance is 0 and disable the button accordingly

const isBalanceZero =

parseFloat(row.balance.replace(",", "")) === 0;

const disabledAttribute = isBalanceZero ? "disabled" : "";

// Use the "render" function to create the "Pay" button for each row

const payButton = `

<button type="button" class="btn btn--primary datatable-btn btn-pay"

data-order="${data}" data-customer="${customerId}"

data-invoiceAmount="${row.grandTotal}" data-balance="${row.balance}"

${disabledAttribute}>

Pay

</button>`;

return payButton;

},

},

],

// Customize the pagination style

lengthMenu: [10, 25, 50, 100], // Number of records to show per page

dom: "<'top'lfr>t<'bottom'ip>", // Layout of pagination elements (l - length changing input, r - processing display element, t - table, i - information summary, p - pagination control)

});

} catch (err) {

console.error(err);

}

});

$(document).on("click", ".view-dr", async function () {

const orderNo = this.getAttribute("data-value");

const response = await fetch(

"./includes/functions/logistics-payment/getDr.php",

{

method: "POST",

body: JSON.stringify({ orderNo: orderNo }),

headers: {

"Content-Type": "application/json",

},

}

);

const data = await response.json();

if (data.length > 0) {

let imgElement = "";

data.forEach((img) => {

imgElement += `<img src="https://buildmart.ph/img/delivery/${img}" class="deliver-pic" />`;

});

Swal.fire({

title: "Delivery Pictures",

html: imgElement,

});

} else {

Swal.fire({

title: "No delivery pictures found!",

icon: "error",

});

}

});

$(document).on("click", ".btn-pay", async function () {

const orderNo = this.getAttribute("data-order");

const customerID = this.getAttribute("data-customer");

const parseBalance = parseFloat(

this.getAttribute("data-balance").replace(/,/g, "")

);

const parseTotalAmount = parseFloat(

this.getAttribute("data-invoiceAmount").replace(/,/g, "")

);

try {

const { value: formValues } = await Swal.fire({

title: "Logistic Payment",

html:

'<label for="swal-input1">Amount:</label>' +

'<input id="swal-input1" class="swal2-input" value="' +

parseBalance +

'">' +

'<label for="swal-input2">Attach File: </label>' +

'<input id="swal-input2" class="swal2-file" type="file">',

focusConfirm: false,

showCancelButton: true,

showLoaderOnConfirm: true,

confirmButtonColor: "#9d0001",

preConfirm: () => {

const payment = parseFloat(

document.getElementById("swal-input1").value

);

if (payment > parseBalance) {

Swal.fire({

title: "Payment is greater than the remaining balance.",

icon: "error",

confirmButtonColor: "#9d0001",

});

return;

} else {

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

window.scrollTo(0, 0);

});

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

const formData = new FormData();

formData.append(

"amount",

document.getElementById("swal-input1").value

);

formData.append("invoiceAmount", parseTotalAmount);

formData.append("orderNo", orderNo);

formData.append("remainingBal", parseBalance);

formData.append("customerId", customerID);

formData.append(

"file",

document.getElementById("swal-input2").files[0]

);

sendData(formData);

}

},

});

} catch (err) {

console.error(err);

}

});

});

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

$(element).load(link, function () {

if (display == 1) console.log(name + " was displayed");

else if (display == 2) alert(name + " was displayed");

});

}

const getCustomers = async () => {

try {

const response = await fetch(

"./includes/functions/logistics-payment/getCustomers.php",

{

method: "GET",

}

);

const data = await response.json();

parentElement.innerHTML =

'<option value="">Select a Logistics Company</option>';

for (customer of data) {

const { id, companyName, owner } = customer;

createCustomerOptions(id, `${companyName} - ${owner}`);

}

} catch (error) {

console.error(error);

}

};

const createCustomerOptions = (id, customerName) => {

const option = document.createElement("option");

option.value = id;

option.textContent = customerName;

parentElement.appendChild(option);

};

async function sendData(formData) {

try {

const response = await fetch(

"./includes/functions/logistics-payment/pay-logistics.php",

{

method: "POST",

body: formData,

}

);

const data = await response.json();

const { success, amountPaid, cwt, netAmount, refNo, balanceAfterPayment } = data;

if (success) {

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

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

Swal.fire({

icon: "info",

title: "Logistic Payment",

html: `Reference No.: ${refNo} <br> Amount Paid: ${amountPaid}`,

});

const remainingBalanceSpan = document.getElementById("remainingBalance");

if (remainingBalanceSpan) {

const formattedBalance = parseFloat(balanceAfterPayment).toLocaleString(undefined, {

minimumFractionDigits: 2,

maximumFractionDigits: 2,

});

remainingBalanceSpan.textContent = formattedBalance;

}

}

} catch (err) {

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

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

Swal.fire({

icon: "error",

title: "Logistic Payment",

text: "Your payment request encountered an error.",

});

}

}

PHP

<?php

include '../dbcon.php';

try {

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

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

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

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

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

$refNo = "RPREF".time();

$date = date("Y/m/d h:m:s");

$balanceAfterPayment = floatVal($remainingBal) - floatVal($amount);

$attachFile = uploadFile('file', $\_FILES);

// Insert to Admin Payment

// $adminSQL = "INSERT INTO `logistic\_admin\_payment`(`customer\_id`,

// `order\_no`, `invoice\_amount`, `bal`, `amount\_paid`)

// VALUES ('$customerId','$orderNo','$invoiceAmount','$balanceAfterPayment','$amount')";

// mysqli\_query($con, $adminSQL);

// Insert to Rpaf Payment

// $balanceAfterPayment = $remainingBal - $invoiceAmount;

$logisticSQL = "INSERT INTO `logistic\_payment`(`reference\_no`, `customer\_id`,

`order\_id`, `logistics\_fee`, `remaining\_balance`, `amount\_paid`, `date`, `file\_attach`) VALUES

('$refNo','$customerId','$orderNo','$logisticsFee','$balanceAfterPayment','$amount',

'$date','$attachFile')";

mysqli\_query($con, $logisticSQL);

echo json\_encode(array(

"success" => true,"amountPaid" => number\_format((double)$amount, 2),

"cwt" => number\_format((double)$cwt, 2),

"netAmount" => number\_format((double)$netAmount, 2),

"refNo" => $refNo,

"balanceAfterPayment" => $balanceAfterPayment)

);

} catch(Exception $e) {

echo $e->getMessage();

}

function uploadFile($id, $file ){

if($file[$id]["name"] != ''){

/\* Getting file name \*/

$filename = time().$file[$id]["name"];

/\* Location \*/

$location = "../../../uploads/logistics-payment/".$filename;

$imageFileType = pathinfo($location,PATHINFO\_EXTENSION);

$imageFileType = strtolower($imageFileType);

$response = $file[$id]["error"];

if(move\_uploaded\_file($file[$id]['tmp\_name'],$location)){

$response = $location;

}

return $filename ;

}

}

?>