<?php

/\*\*

 \* Plugin Name: PWA Camera

 \* Plugin URI: https://www.pwa.com/

 \* Description: PWA Camera Plugin

 \* Version: 1.0.11

 \* Author: Alfred

 \* Author URI: http://www.pwa.com

 \*/

/\*

v1.0.9

- email configuration

- shortcode pwa camera with device selection

- support environment facing camera

v1.0.10

- bugs fixes on syntax return

\*/

/\*\*\*\*\*\*\*\*\*\*\* ADMIN MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

add\_action('admin\_menu', 'setup\_menu');

//for add/edit records

//add\_action('admin\_post\_nopriv\_pc-add', 'pcAdd');

//add\_action('admin\_post\_pc-add', 'pcAdd');

//for delete records

add\_action('admin\_post\_nopriv\_pc-delete', 'pcDelete');

add\_action('admin\_post\_pc-delete', 'pcDelete');

//action: save

add\_action('admin\_post\_nopriv\_save', 'save');

add\_action('admin\_post\_save', 'save');

// register shortcode

add\_shortcode('pwacamera', 'camera\_shortcode');

//menu, submenu init

function setup\_menu(){

 //page title, menu label, ?, query parameter, callback

 add\_menu\_page( 'Camera', 'Camera', 'manage\_options', 'init', 'init' );

 //admin page without menu - view/edit member, delete member page

 //add\_submenu\_page(null, 'View/Edit Package Calculator Record', 'View/Edit Package Calculator Record', 'manage\_options', 'pc-edit', 'pcEdit');

 add\_submenu\_page(null, 'Delete Package Calculator Record', 'Delete Package Calculator Record', 'manage\_options', 'pc-delete', 'pcDelete');

}

//init menu

function init(){

 //db for email configuration

 echo "<script>

 function doValidate(){

 if(document.getElementById('emailAddr').value.trim()==''){

 alert('Email Address is required!');

 return false;

 } else {

 if (/^[a-zA-Z0-9.!#$%&'\*+/=?^\_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)\*$/.test(document.getElementById('emailAddr').value)){

 return true;

 } else {

 alert('Invalid Email Address!');

 }

 return false;

 }

 }

 </script>";

 //check if email config table exists?

 global $wpdb;

 $sqlTableHeader = 'CREATE TABLE `'.$wpdb->prefix.'pwa\_email` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `email` varchar(500) COLLATE utf8mb4\_unicode\_ci NOT NULL,

 `createddate` datetime DEFAULT NULL,

 `lastupdated` datetime DEFAULT NULL,

 UNIQUE KEY `ID` (`id`)

 ) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;';

 require\_once( ABSPATH . 'wp-admin/includes/upgrade.php' );

 dbDelta( $sqlTableHeader );

 global $wpdb;

 $result = $wpdb->get\_results ("SELECT \* FROM " . $wpdb->prefix . "pwa\_email ORDER BY id DESC LIMIT 1"); //latest on top

 $rowCount = $wpdb->num\_rows;

 $email="";

 if($rowCount>0){

 foreach ( $result as $row ){

 $email = $row->email;

 }

 }

 //check if table exists?

 global $wpdb;

 $sqlTableHeader = 'CREATE TABLE `'.$wpdb->prefix.'pwa\_camera` (

 `id` int(11) NOT NULL AUTO\_INCREMENT,

 `imagefile` varchar(500) COLLATE utf8mb4\_unicode\_ci NOT NULL,

 `createddate` datetime DEFAULT NULL,

 `lastupdated` datetime DEFAULT NULL,

 UNIQUE KEY `ID` (`id`)

 ) ENGINE=InnoDB AUTO\_INCREMENT=0 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;';

 require\_once( ABSPATH . 'wp-admin/includes/upgrade.php' );

 dbDelta( $sqlTableHeader );

 echo "<h1>Email Setting</h1>";

 if(isset($\_GET["act"])){

 if($\_GET["act"]=="1"){

 echo '<div class="notice notice-warning is-dismissible"><p>Email address is saved successfully!</p></div>';

 }

 }

 echo "<p>Configure your email address here to received the PWA Image Capture:</p>";

 echo "<form id='emailForm' name='emailForm' method='POST' action='". esc\_url(admin\_url('admin-post.php')) . "?action=update' onsubmit='return doValidate();'>";

 echo "<table cellpadding='2'>";

 echo '<tr><td>\*Email Address:</td><td><input type="text" id="emailAddr" name="emailAddr" class="" style="width:300px;" value="'.$email.'"/></td></tr>';

 echo "<tr><td>&nbsp;</td><td></td></tr></table>";

 echo "<input type='submit' value='Save' class='button-primary' style='width:120px;height:40px;'/>";

 echo "<p>&nbsp;</p><hr /></p>&nbsp;</p>";

 echo "<h1>Captured Image Listing</h1>";

 global $wpdb;

 $result = $wpdb->get\_results ("SELECT \* FROM " . $wpdb->prefix . "pwa\_camera ORDER BY createddate DESC"); //latest on top

 $rowCount = $wpdb->num\_rows;

 $imagefile="";

 $createddate="";

 $id=-1;

 $short\_description = "";

 if($rowCount>0){

 foreach ( $result as $row ){

 $id = $row->id;

 $imagefile = $row->imagefile;

 $createddate = $row->createddate;

 $imURL = '../wp-content/uploads/'.$imagefile;

 echo '<a href="'.$imURL.'" target="\_blank">'.$imagefile . '</a> Taken: ' . $createddate . '<br />';

 }

 } else {

 echo 'No captured image found.';

 }

 jsDelete(); //print delete js

}

/\*admin function to update email\*/

add\_action( 'admin\_post\_nopriv\_update', 'update' );

add\_action( 'admin\_post\_update', 'update');

function update(){

 //template\_tour\_id

 //template\_price\_id

 //selectedValues

 if(isset($\_GET["action"])){

 if($\_GET["action"]=="update"){

 $email = $\_POST["emailAddr"];

 global $wpdb;

 $wpdb->insert(

 $wpdb->prefix.'pwa\_email',

 array(

 'email' => $email,

 'lastupdated' => current\_time('mysql', 1),

 'createddate' => current\_time('mysql', 1)

 )

 );

 }

 wp\_redirect(admin\_url('admin.php?page=init&act=1'));

 }

}

/\*end of admin function to update email\*/

/\*\*\*\*\*\*\*\* steps and feature table \*\*\*\*\*\*\*\*\*\*\*/

function jsDelete(){

 echo '<script>

 function confirmDelete(tbl, id){

 if(confirm("Are you sure want to delete this record?")){

 document.location.href="'.esc\_url(admin\_url('admin-post.php')).'?action=pc-delete&tbl=" + tbl + "&id=" + id;

 }

 }

 </script>';

}

/\*\*\*\*\*\*\* for record DELETE \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

function pcDelete(){

 if(isset($\_GET["action"])){

 if($\_GET["action"]=="pc-delete"){

 $tbl = "";

 if(isset($\_GET["tbl"])){

 if($\_GET["tbl"]!=""){

 $tbl = $\_GET["tbl"];

 //$id = -1;

 //if(isset($\_POST[]))

 switch ($tbl) {

 case "pwa\_camera":

 dbPackage();

 break;

 default:

 echo '<h1>ERROR: Invalid Action</h1>';

 }

 }

 } else {

 echo '<h1>ERROR: Invalid Action</h1>';

 }

 }

 } else {

 echo '<h1>ERROR: Invalid Action</h1>';

 }

}

/\*\*\*\*\*\*\*\*\*\*\* SHORT CODE \*\*\*\*\*\*\*\*\*\*\*\*\*/

function camera\_shortcode(){

 ob\_start();

 //echo style

 echo '<style>

 #container {

 margin: 0px auto;

 width: 500px;

 height: 375px;

 border: 10px #333 solid;

 }

 #videoElement {

 width: 500px;

 height: 375px;

 background-color: #666;

 }

 #canvas{

 display:none;

 }

 .output{

 display:none;

 }

 </style>';

 //echo html + js

echo '<center><h2><a id="lblAction" href="javascript:;" onclick="toggleCamera();">Open Camera</a></h2></center>

<div id="container">

 <video controls autoplay playsInline id="videoElement">

 </video>

</div>

<center><h2><a id="lblAction" href="javascript:;" onclick="takePicture();">&#128247;</a></h2></center>

<center>v1.0.11</center>

<canvas id="canvas" style="display:none;">

</canvas>

<div class="output" style="display:none;">

 <img id="photo" alt="The screen capture will appear in this box.">

</div>

<a href="" id="downloadPhoto" download="my-photo.jpg" style="display:none;">Download Photo</a>

<script>

//var width = 1280; // We will scale the photo width to this

//var height = 720; // This will be computed based on the input stream

var width = 420; // We will scale the photo width to this

var height = 360;

 var streaming = false;

var canvas = null;

var photo = null;

var startbutton = null;

var video = document.querySelector("#videoElement");

canvas = document.getElementById("canvas");

photo = document.getElementById("photo");

const supports = navigator.mediaDevices.getSupportedConstraints();

 if (!supports["facingMode"]) {

 alert("Browser Not supported!");

 }

let stream;

 const capture = async facingMode => {

 const options = {

 audio: false,

 video: {

 facingMode,

 },

 };

 try {

 if (stream) {

 const tracks = stream.getTracks();

 tracks.forEach(track => track.stop());

 height = video.videoHeight / (video.videoWidth/width);

 video.setAttribute("width", width);

 video.setAttribute("height", height);

 canvas.setAttribute("width", width);

 canvas.setAttribute("height", height);

 }

 stream = await navigator.mediaDevices.getUserMedia(options);

 } catch (e) {

 alert(e);

 return;

 }

 video.srcObject = null;

 video.srcObject = stream;

 video.play();

 }

function toggleCamera(e){

 if(document.getElementById("lblAction").innerHTML=="Open Camera"){

 document.getElementById("lblAction").innerHTML = "Close Camera";

 capture("environment");

 } else {

 document.getElementById("lblAction").innerHTML = "Open Camera";

 stop();

 }

}

function stop(e) {

 var stream = video.srcObject;

 var tracks = stream.getTracks();

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

 var track = tracks[i];

 track.stop();

 }

 //video.pause();

 video.srcObject = null;

}

function takePicture() {

 video.pause();

 var context = canvas.getContext("2d");

 console.log(width, height);

 console.log((width && height));

 if (width && height) {

 console.log("take picture");

 canvas.width = width;

 canvas.height = height;

 context.drawImage(video, 0, 0, width, height);

 var data = canvas.toDataURL("image/jpeg", 1.0);

 photo.setAttribute("src", data);

 //console.log(data);

 saveImage(data);

 downloadImage();

 //video.pause();

 //stop(); //close camera

 //document.getElementById("lblAction").innerHTML = "Open Camera";

 document.getElementById("downloadPhoto").click();

 setTimeout(function () {

 video.play();

 }, 700);

 } else {

 clearphoto();

 }

}

 //download image

 function downloadImage() {

 const download = document.getElementById("downloadPhoto");

 const canvas = document.getElementById("canvas").toDataURL("image/jpeg").replace("image/jpeg", "image/octet-stream");

 download.setAttribute("href", canvas);

 //download.style.display="block";

}

 //end download image

 function clearphoto() {

 var context = canvas.getContext("2d");

 context.fillStyle = "#AAA";

 context.fillRect(0, 0, canvas.width, canvas.height);

 var data = canvas.toDataURL("image/jpeg", 1.0);

 photo.setAttribute("src", data);

 }

 clearphoto();

function saveImage(data){

 const formData = new FormData();

 formData.append("dataURL", data);

 fetch("'.esc\_url(admin\_url("admin-post.php")) . '?action=save'.'", {

 method: "POST",

 body: formData

 });

 //alert("Photo is Taken.");

}

</script>';

 //end echo html + js

 return ob\_get\_clean();

}

function save(){

 if(isset($\_GET["action"])){

 if($\_GET["action"]=="save"){

 $data = $\_POST["dataURL"];

 //$data = 'data:image/png;base64,AAAFBfj42Pj4';

 list($type, $data) = explode(';', $data);

 list(, $data) = explode(',', $data);

 $data = base64\_decode($data);

 $imFile = 'image-'.date('YmdHis').'.jpg';

 file\_put\_contents('../wp-content/uploads/'.$imFile, $data);

 global $wpdb;

 $wpdb->insert(

 $wpdb->prefix.'pwa\_camera',

 array(

 'imagefile' => $imFile,

 'lastupdated' => current\_time('mysql', 1),

 'createddate' => current\_time('mysql', 1)

 )

 );

 $record\_id = $wpdb->insert\_id;

 }

 }

}

?>