<%- include("../../views/partials/admin/header") %>

<head>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/cropperjs/dist/cropper.min.css">

</head>

<style>

.error-message {

color: red;

margin-top: 5px;

font-size: 14px;

}

.card {

position: relative;

display: flex;

flex-direction: column;

min-width: 0;

word-wrap: break-word;

background-color: #fff;

background-clip: border-box;

border: 0 solid transparent;

border-radius: .25rem;

margin-bottom: 1.5rem;

box-shadow: 0 2px 6px 0 rgb(218 218 253 / 65%), 0 2px 6px 0 rgb(206 206 238 / 54%);

}

.me-2 {

margin-right: .5rem!important;

}

*@import* url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap');

:root {

--blue: #0071FF;

--light-blue: #B6DBF6;

--dark-blue: #005DD1;

--grey: #f2f2f2;

}

body {

background: var(--light-blue);

}

.img-area .icon {

font-size: 100px;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

.img-area h3 {

font-size: 20px;

font-weight: 500;

margin-bottom: 6px;

}

.img-area {

position: relative;

width: 100%;

height: 240px;

background: var(--grey);

margin-bottom: 30px;

border-radius: 15px;

overflow: hidden;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

}

.img-area p {

color: #999;

}

.img-area p span {

font-weight: 600;

}

.img-area img {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover;

object-position: center;

z-index: 100;

}

.select-image {

display: block;

width: 100%;

padding: 16px 0;

border-radius: 15px;

background: var(--blue);

color: #fff;

font-weight: 500;

font-size: 16px;

border: none;

cursor: pointer;

transition: all .3s ease;

}

.thumbnails-container {

display: flex;

overflow-x: auto;

}

.thumbnail {

margin-right: 10px;

}

</style>

<section class="content-main">

<div class="row">

<div class="col-9">

<div class="content-header">

<h2 class="content-title">Add New Banner</h2>

</div>

</div>

<div class="col-lg-6">

<div class="card mb-4">

<div class="card-body">

<form method="" action="" enctype="multipart/form-data"

onsubmit="*return* validateForm()">

<div class="mb-4">

<label for="product\_name" class="form-label">Banner Title</label>

<input type="text" placeholder="Type here" name="title" class="form-control border"

id="product\_name">

<div id="productName-error" class="error-message"></div>

</div>

<div class="mb-4">

<label class="form-label">Description</label>

<textarea placeholder="Type here" name="description" class="form-control border"

rows="4"></textarea>

<div id="description" class="error-message"></div>

</div>

<div class="mb-4">

<label for="startingDate" class="form-label">Start Date</label>

<input type="date" name="startDate" class="form-control" required id="startingDate" />

<div id="error-start-date" class="error-message"></div>

</div>

<div class="mb-4">

<label for="expiringDate" class="form-label">End Date</label>

<input type="date" name="endDate" class="form-control" id="expiringDate" required />

<div id="error-end-date" class="error-message"></div>

</div>

<div class="col-lg-4">

<div class="mb-4">

<label class="form-label">Link</label>

<input placeholder="" name="link" type="text" class="form-control border">

<div id="link-error" class="error-message"></div>

</div>

</div>

<div class="card-header">

<h4>Choose Banner Image</h4>

</div>

<div class="border row">

<div id="addedImagesContainer" class="thumbnails-container"></div>

</div>

<div class="card-body">

<img src="user/imgs/theme/upload.svg" alt="" id="imgView1">

<input class="form-control" type="file" name="images" id="inputFile"

accept="image/png, image/jpeg, image/jpg"

onchange="viewImage1(event),viewImage(event, 1)">

<div id="image-error" class="error-message"></div>

</div>

<div class="image-cropper" style="display:none; width: 300px; height: 200px;">

<img src="" id="croppedImg1" alt="">

<button type="button" id="saveButton1" class="btn-sm btn-primary">Save</button>

</div>

</div>

<div>

<button class="btn btn-md rounded font-sm hover-up ml-200" type="button"

onclick="validateAndSubmit()">Publish</button>

</div>

</form>

</div>

</div>

</div>

</div>

</section>

<script src="https://cdn.jsdelivr.net/npm/cropperjs/dist/cropper.min.js"></script>

<script>

const selectedImages = [];

document.getElementById("imageInput").addEventListener("change", handleFileSelect);

function handleFileSelect(*event*) {

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

addedImagesContainer.innerHTML = "";

const files = *event*.target.files;

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

const file = files[i];

selectedImages.push(file);

const thumbnail = document.createElement("div");

thumbnail.classList.add("thumbnail");

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

img.src = URL.createObjectURL(file);

img.alt = "thumbnail";

img.style.width = "60px";

img.style.height = "auto";

const removeIcon = document.createElement("span");

removeIcon.classList.add("remove-icon");

removeIcon.innerHTML = "&times;";

removeIcon.addEventListener("click", function () {

const index = selectedImages.indexOf(file);

*if* (index !== -1) {

selectedImages.splice(index, 1);

}

thumbnail.remove();

});

thumbnail.appendChild(img);

thumbnail.appendChild(removeIcon);

addedImagesContainer.appendChild(thumbnail);

}

}

document.addEventListener("DOMContentLoaded", function () {

});

function validateAndSubmit() {

*if* (validateForm()) {

document.forms[0].submit();

}

}

function validateForm() {

resetErrorMessages();

var isValid = true;

var productName = document.getElementById("product\_name").value;

const sDate = document.getElementsByName('startDate')[0].value;

const eDate = document.getElementsByName('endDate')[0].value;

*if* (!sDate) {

displayErrorMessage("error-start-date", 'Starting date is required');

isValid = false;

}

*if* (!eDate) {

displayErrorMessage("error-end-date", 'End date is required');

isValid = false;

}

const sDateObj = new Date(sDate);

const eDateObj = new Date(eDate);

const todayDateObj = new Date();

todayDateObj.setHours(0, 0, 0, 0);

*if* (sDateObj > eDateObj) {

document.getElementById('error-end-date').innerHTML = 'End date should be after the start date';

isValid = false;

}

*if* (productName === "") {

displayErrorMessage("productName-error", "Product Name is required");

isValid = false;

}

var description = document.getElementsByName('description')[0].value;

*if* (description === "") {

displayErrorMessage("description-error", "Description is required");

isValid = false;

}

var link = document.getElementsByName('link')[0].value;

*if* (link === "") {

displayErrorMessage("link-error", "Link is required");

isValid = false;

}

let imageFile=document.getElementById('inputFile');

let File=imageFile.files[0]

console.log(File,"File");

*if* (File===undefined) {

console.log('Invalid file type. Please upload a JPEG, PNG, or GIF file.');

displayErrorMessage("image-error", "Invalid file type. Please upload a JPEG, PNG, or GIF file.");

isValid = false;

}

*return* isValid;

}

function displayErrorMessage(*elementId*, *message*) {

var errorElement = document.getElementById(*elementId*);

errorElement.innerText = *message*;

errorElement.style.display = "block";

}

function resetErrorMessages() {

var errorElements = document.querySelectorAll(".error-message");

errorElements.forEach(function (*element*) {

*element*.innerText = "";

*element*.style.display = "none";

});

}

function viewImage1(*event*) {

document.getElementById('imgView1').src = URL.createObjectURL(*event*.target.files[0])

}

function viewImage(*event*, *index*) {

let input = *event*.target;

let reader = new FileReader();

reader.onload = function () {

let dataURL = reader.result;

let image = document.getElementById('imgView' + *index*);

image.src = dataURL;

let cropper = new Cropper(image, {

aspectRatio:1360 / 777,

viewMode: 1,

guides: true,

background: false,

autoCropArea: 0.8,

zoomable: true

});

let cropperContainer = document.querySelector('#croppedImg' + *index*).parentNode;

cropperContainer.style.display = 'block';

let saveButton = document.querySelector('#saveButton' + *index*);

saveButton.addEventListener('click', async function () {

let croppedCanvas = cropper.getCroppedCanvas();

let croppedImage = document.getElementById("croppedImg" + *index*);

croppedImage.src = croppedCanvas.toDataURL('image/jpeg', 1.0);

let timestamp = new Date().getTime();

let fileName = `cropped-img-${timestamp}-${*index*}.png`;

*await* croppedCanvas.toBlob(*blob* => {

let input = document.getElementById('input' + *index*)

let imgFile = new File([*blob*], fileName, *blob*)

const fileList = new DataTransfer();

fileList.items.add(imgFile);

input.files = fileList.files

});

cropperContainer.style.display = 'none';

cropper.destroy();

});

};

reader.readAsDataURL(input.files[0]);

}

</script>

<%- include("../../views/partials/admin/footer") %>