User add property

@model ClassLibrary.WebModels.AddUserPropertyModel

@{

ViewBag.Title = "addproperty";

Layout = "~/Views/Shared/\_UserLayout.cshtml";

var Data = new EntitiesLibrary.Data.Registration((EntitiesLibrary.tRegistration)Session["AccountDetails"]);

}

<section class="inner-pages">

<div class="container">

<div class="row">

<div class="col-md-12">

<div class="innr-pg-ttle">

Add a Property

</div><!-- /innr-pg-ttle -->

</div><!-- /col-md-12 -->

</div><!-- /row -->

<br>

<div class="row">

@using (Ajax.BeginForm("addproperty", "User", null, new AjaxOptions

{

HttpMethod = "POST",

OnSuccess = "addpropertySuccess",

OnFailure = "ajaxFailure",

}, new { id = "user-add-property-form" }))

{

@Html.HiddenFor(m => m.Latitude)

@Html.HiddenFor(m => m.PropertyImagesString)

@Html.HiddenFor(m => m.IsJob, new { Value = false })

@Html.HiddenFor(m => m.Longitude)

@Html.HiddenFor(m => m.JobType, new { Value = ClassLibrary.Enum.JobType.Business })

@Html.HiddenFor(m => m.PropertyType, new { Value = ClassLibrary.Enum.PropertyType.Airbnb })

@Html.HiddenFor(m => m.UserId, new { Value = (Data == null ? "0" : Data.UserId) })

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

<div class="personal-details">

<img src="~/Content/images/icons/add-property.png" />

<span>Add a property</span>

</div><!-- /personal-details -->

<div class="row">

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

<div class="form-group">

<input type="radio" name="sp" value="@ClassLibrary.Enum.JobType.Business" class="radio-jobtype" id="radio-jobtype-Business" checked />

@\*<input type="radio" id="test1" name="sp" />\*@

<label for="radio-jobtype-Business">Business</label>

</div>

</div>

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

<div class="form-group">

<input type="radio" name="sp" value="@ClassLibrary.Enum.JobType.Residential" class="radio-jobtype" id="radio-jobtype-Residential" />

@\*<input type="radio" id="test2" name="sp" />\*@

<label for="radio-jobtype-Residential">Residential</label>

</div>

</div>

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.PropertyName, new { autocomplete = "off", placeholder = "Property Name", Name = "PropertyName", @class = "form-control" })

<span id="spn-PropertyName" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Property Name" />\*@

</div>

<h4 class="form-title">Details</h4>

<div class="form-group">

@Html.TextBoxFor(m => m.FirstName, new { autocomplete = "off", placeholder = "First Name", Name = "firstname", @class = "form-control" })

<span id="spn-firstname" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="First Name" />\*@

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.LastName, new { autocomplete = "off", placeholder = "Last Name", Name = "lastname", @class = "form-control" })

<span id="spn-lastname" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Last Name" />\*@

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.Phone, new { autocomplete = "off", placeholder = "Phone Number", Name = "phone", @class = "form-control" })

<span id="spn-phone" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Phone Number" />\*@

</div>

</div><!-- /col-md-4 -->

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

<div class="personal-details-no-icon">

<span>Address Details</span>

</div><!-- /personal-details -->

<div class="form-group">

@Html.TextBoxFor(m => m.Address, new { autocomplete = "off", placeholder = "Address", Name = "Address", @class = "form-control" })

<span id="spn-Address" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Address" />\*@

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.Suburb, new { autocomplete = "off", placeholder = "Suburb", Name = "Suburb", @class = "form-control" })

<span id="spn-Suburb" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Suburb" />\*@

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.State, new { autocomplete = "off", placeholder = "State", Name = "State", @class = "form-control" })

<span id="spn-State" style="color:red;display:none;"></span>

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.Country, new { autocomplete = "off", placeholder = "Country", Name = "Country", @class = "form-control", Value = "Australia", @readonly = "readonly" })

<span id="spn-Country" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" value="Australia" />\*@<!-- Dynamic -->

</div>

<div class="form-pad"></div>

<div class="personal-details-no-icon">

<span>Add photos</span>

</div><!-- /personal-details -->

<ul class="upload-imgs">

<li>

<input type="file" class="my\_file" style="display:none;" />

<input type="image" class="user-property-images" style="max-width: 76px;" src="~/Content/images/add-images.png" />

<input type="hidden" class="hfn-base64" />

</li>

<li>

<input type="file" class="my\_file" style="display:none;" />

<input type="image" class="user-property-images" style="max-width: 76px;" src="~/Content/images/add-images.png" />

<input type="hidden" class="hfn-base64" />

</li>

<li>

<input type="file" class="my\_file" style="display:none;" />

<input type="image" class="user-property-images" style="max-width: 76px;" src="~/Content/images/add-images.png" />

<input type="hidden" class="hfn-base64" />

</li>

<li>

<input type="file" class="my\_file" style="display:none;" />

<input type="image" class="user-property-images" style="max-width: 76px;" src="~/Content/images/add-images.png" />

<input type="hidden" class="hfn-base64" />

</li>

</ul>

</div><!-- /col-md-4 -->

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

<div class="personal-details-no-icon">

<span>Type</span>

</div><!-- /personal-details -->

<div class="form-group">

<select class="form-control" id="user-add-selct-propertytype">

<option value="@ClassLibrary.Enum.PropertyType.Airbnb.ToString()">Airbnb</option>

<option value="@ClassLibrary.Enum.PropertyType.Domestic.ToString()">Domestic</option>

<option value="@ClassLibrary.Enum.PropertyType.Commercial.ToString()">Commercial</option>

</select>

</div>

<div class="form-pad"></div>

<div class="personal-details-no-icon">

<span>Property Details</span>

</div><!-- /personal-details -->

<div class="form-group">

@Html.TextBoxFor(m => m.BedroomNum, new { autocomplete = "off", placeholder = "Bedrooms", Name = "BedroomNum", @class = "form-control" })

<span id="spn-BedroomNum" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Bedrooms" />\*@

</div>

<div class="form-group">

@Html.TextBoxFor(m => m.BathroomNum, new { autocomplete = "off", placeholder = "Bathrooms", Name = "BathroomNum", @class = "form-control" })

<span id="spn-BathroomNum" style="color:red;display:none;"></span>

@\*<input type="text" class="form-control" placeholder="Bathrooms" />\*@

</div>

<div class="form-group">

@Html.TextAreaFor(m => m.Description, new { autocomplete = "off", placeholder = "Description", Name = "Description", @class = "form-control", rows = "3" })

<span id="spn-Description" style="color:red;display:none;"></span>

@\*<textarea rows="3" class="form-control" placeholder="Description"></textarea>\*@

</div>

</div><!-- /col-md-4 -->

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">

<div class="row">

<div class="col-md-4 col-md-offset-4">

<div class="form-pad"></div>

<input type="button" value="Add" id="btn-dummy-propertyadd" class="btn btn-default view-job-btn view-btn" style="width:100%;" />

<input type="submit" value="Add" id="btn-propertyadd" class="btn btn-default view-job-btn view-btn" style="width:100%;display:none;" />

</div>

</div>

</div>

}

</div><!-- /row -->

<div class="spacing"></div>

<br>

</div><!-- /container -->

</section><!-- /main-banner -->

<br><br><br><br>

<script src="https://maps.googleapis.com/maps/api/js?libraries=places&key=AIzaSyCR61VJlK15kYWdqQnUjyeAqylMYkugnfQ"></script>

<script>

$(document).on('click', ".user-property-images", function (e) {

$(this).parent().find('.my\_file').click();

e.preventDefault();

//$("#imageUpload").click();

});

function previewPropertyImage(uploader, source) {

//ensure a file was selected

if (uploader.files && uploader.files[0]) {

var imageFile = uploader.files[0];

var fileType = imageFile["type"];

var ValidImageTypes = ["image/jpeg", "image/png"];

if ($.inArray(fileType, ValidImageTypes) < 0) {

toastr.error("Invalid file type");

}

else {

var reader = new FileReader();

reader.onload = function (e) {

//set the image data as source

$(source).parent().find('.user-property-images').attr('src', e.target.result);

//console.log($(source).parent().find('.user-property-images').data('src'));

$(source).parent().find('.hfn-base64').val(e.target.result);

//$('#ProfilePic').val(e.target.result);

}

reader.readAsDataURL(imageFile);

}

}

}

$(document).on('change', ".my\_file", function () {

previewPropertyImage(this, $(this));

});

</script>

<script>

$(document).on('click', '.radio-jobtype', function () {

var source = $(this);

$('#JobType').val($(source).val());

});

$(document).on('change', '#user-add-selct-propertytype', function () {

var source = $(this);

var propertyTYpe = $(this).find(":selected").val();

$('#user-add-selct-propertytype').val(propertyTYpe);

$('#PropertyType').val(propertyTYpe);

});

$(document).on('keydown', "#Phone , #BedroomNum ,#BathroomNum", function (e) {

// Allow: backspace, delete, tab, escape, enter and .

if ($.inArray(e.keyCode, [46, 8, 9, 27, 13, 110, 190]) !== -1 ||

// Allow: Ctrl+A, Command+A

(e.keyCode === 65 && (e.ctrlKey === true || e.metaKey === true)) ||

// Allow: home, end, left, right, down, up

(e.keyCode >= 35 && e.keyCode <= 40)) {

// let it happen, don't do anything

return;

}

// Ensure that it is a number and stop the keypress

if ((e.shiftKey || (e.keyCode < 48 || e.keyCode > 57)) && (e.keyCode < 96 || e.keyCode > 105)) {

e.preventDefault();

}

});

function addpropertySuccess(data) {

$(".preloder").fadeOut();

if (data.Status) {

toastr.success("Property added");

setTimeout(function () {

window.location.href = "/User/myproperty";

}, 500);

}

else {

$('#btn-dummy-propertyadd').prop('disabled', false);

toastr.error(data.Message);

}

}

$(document).on('click', '#btn-dummy-propertyadd', function () {

var status = true;

$('#user-add-property-form input[type="text"], TextArea').each(function () {

if (this.name != 'PropertyName') {

if ($(this).val() != null && $(this).val() != "") {

$(this).parent().find('#spn-' + this.name).hide();

}

else {

$(this).parent().find('#spn-' + this.name).text("Required").show();

status = false;

}

}

});

setTimeout(function () {

if (status) {

var obj = new Object();

obj.PropertyImagesList = [];

$('.user-property-images').each(function () {

var currentElement = $(this);

var value = $(currentElement).parent().find('.hfn-base64').val();

if (value != null && value != "") {

var newobj = new Object();

newobj.base64 = value;

newobj.fileId = "";

obj.PropertyImagesList.push(newobj);

}

});

//obj.PropertyName = $('#PropertyName').val();

//obj.FirstName = $('#FirstName').val();

//obj.LastName = $('#LastName').val();

//obj.Phone = $('#Phone').val();

//obj.Address = $('#Address').val();

//obj.Suburb = $('#Suburb').val();

//obj.State = $('#State').val();

//obj.Country = $('#Country').val();

//obj.BedroomNum = $('#BedroomNum').val();

//obj.BathroomNum = $('#BathroomNum').val();

//obj.Description = $('#Description').val();

//obj.IsJob = false;

//obj.Latitude = $('#Latitude').val();

//obj.Longitude = $('#Longitude').val();

//obj.JobType = $('#JobType').val();

//obj.PropertyType = $('#PropertyType').val();

$('#PropertyImagesString').val(JSON.stringify(obj));

$('#btn-dummy-propertyadd').prop('disabled', true);

$('#btn-propertyadd').trigger('click');

$(".preloder").show();

}

}, 300);

});

</script>

<script>

var componentForm = {

street\_number: 'long\_name',

route: 'long\_name',

locality: 'long\_name',

administrative\_area\_level\_1: 'long\_name',

country: 'long\_name',

postal\_code: 'long\_name',

neighborhood: 'long\_name',

sublocality\_level\_3: 'long\_name',

sublocality\_level\_2: 'long\_name',

sublocality\_level\_1: 'long\_name'

};

google.maps.event.addDomListener(window, 'load', function () {

var places = new google.maps.places.Autocomplete(document.getElementById('Address'), { componentRestrictions: { country: 'au' } });

var cities = new google.maps.places.Autocomplete(document.getElementById('Suburb'), { types: ['(cities)'], componentRestrictions: { country: 'au' } });

var state = new google.maps.places.Autocomplete(document.getElementById('State'), { types: ['(regions)'], componentRestrictions: { country: 'au' } });

//var country = new google.maps.places.Autocomplete(document.getElementById('Country'), { types: ['(regions)'], componentRestrictions: { country: 'au' } });

google.maps.event.addListener(cities, 'place\_changed', function () {

var cityFieldValue = $('#Suburb').val();

cityFieldValue = cityFieldValue.substring(0, cityFieldValue.indexOf(','));

$('#Suburb').val(cityFieldValue);

});

google.maps.event.addListener(state, 'place\_changed', function () {

var stateFieldValue = $('#State').val();

stateFieldValue = stateFieldValue.substring(0, stateFieldValue.indexOf(','));

$('#State').val(stateFieldValue);

});

//google.maps.event.addListener(country, 'place\_changed', function () {

// var countryFieldValue = $('#Country').val();

// $('#Country').val(countryFieldValue);

//});

google.maps.event.addListener(places, 'place\_changed', function () {

var addressFieldValue = $('#Address').val();

//$('#city').val('');

//$('#StreetNumber').val('');

//$('#country').val('');

//$('#postalCode').val('');

//$('#Route').val('');

//$('#UserLat').val('');

//$('#UserLong').val('');

var place = places.getPlace();

var address = place.formatted\_address;

var latitude = place.geometry.location.lat();

var longitude = place.geometry.location.lng();

addressFieldValue = addressFieldValue.substring(0, addressFieldValue.indexOf(','));

for (var i = 0; i < place.address\_components.length; i++) {

var addressType = place.address\_components[i].types[0];

//console.log(addressType);

if (addressType == "locality") {

var val = place.address\_components[i][componentForm[addressType]];

//console.log(val)

$('#Suburb').val(val);

}

if (addressType == "route") {

var val = place.address\_components[i][componentForm[addressType]];

//console.log(val)

}

//if (addressType == "postal\_code") {

// var val = place.address\_components[i][componentForm[addressType]];

// $('#Zip').val(val);

// //console.log(val)

//}

if (addressType == "neighborhood") {

var val = place.address\_components[i][componentForm[addressType]];

//console.log(val)

}

if (addressType == "administrative\_area\_level\_1") {

var val = place.address\_components[i][componentForm[addressType]];

//console.log(val)

$('#State').val(val);

}

//if (addressType == "country") {

// var val = place.address\_components[i][componentForm[addressType]];

// //console.log(val)

// $('#Country').val(val);

//}

if (addressType == "street\_number") {

var val = place.address\_components[i][componentForm[addressType]];

//console.log(val)

//$('#StreetNumber').val(val);

}

}

$('#Address').val(addressFieldValue);

$('#Latitude').val(latitude);

$('#Longitude').val(longitude);

//var mesg = "Address: " + address;

//mesg += "\nLatitude: " + latitude;

//mesg += "\nLongitude: " + longitude;

});

});

</script>