User post job view

@model ClassLibrary.WebModels.PostJobModel

@{

ViewBag.Title = "postjob";

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">

Post Jobs

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

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

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

<br>

<div class="row">

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

{

HttpMethod = "POST",

OnSuccess = "postjobSuccess",

OnFailure = "ajaxFailure",

}, new { id = "user-post-job-form" }))

{

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

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

@Html.HiddenFor(m => m.IsNewProperty, new { Value = "true" })

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

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

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

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

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

@Html.HiddenFor(m => m.CleanerId, new { Value ="0" })

@Html.HiddenFor(m => m.PropertyId, new { Value = "0" })

@Html.HiddenFor(m => m.TimeZone, new { Value = "UTC" })

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

<div class="personal-details">

<img src="~/Content/images/icons/post-job.png" />

<span>Post a job</span>

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

<div class="form-group">

<select class="form-control" id="slct-user-property">

<option value="0">Select my Property</option>

@if (Data != null)

{

if (Data.UserProperties.Count > 0)

{

foreach (var property in Data.UserProperties)

{

<option value="@property.PropertyId">@(property.PropertyName ?? string.Empty)</option>

}

}

}

</select>

</div>

<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 />

<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" />

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

</div>

</div>

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

<div class="form-group">

<input type="radio" name="sp1" value="@ClassLibrary.Enum.WorkType.Cleaning" class="radio-worktype" id="radio-worktype-Cleaning" checked />

<label for="radio-worktype-Cleaning">Cleaning</label>

</div>

</div>

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

<div class="form-group">

<input type="radio" name="sp1" value="@ClassLibrary.Enum.WorkType.MeetORgreet" class="radio-worktype" id="radio-worktype-Meet\_Greet" />

<label for="radio-worktype-Meet\_Greet">Meet/Greet</label>

</div>

</div>

</div>

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="Property Name" name="prpertyname" id="user-job-prpertyname" />\*@

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

</div>

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

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="First Name" name="firstname" id="user-job-firstname" />\*@

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

</div>

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="Last Name" name="lastname" id="user-job-lastname" />\*@

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

</div>

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="Phone Number" name="phone" id="user-job-phone" />\*@

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

</div>

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

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

<span>Property Images</span>

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

<ul class="upload-imgs user-property-images-ul">

<li>

<input type="file" class="my\_file1" 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" data-fileid="" data-filepath="" />

</li>

<li>

<input type="file" class="my\_file1" 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" data-fileid="" data-filepath="" />

</li>

<li>

<input type="file" class="my\_file1" 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" data-fileid="" data-filepath="" />

</li>

<li>

<input type="file" class="my\_file1" 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" data-fileid="" data-filepath="" />

</li>

</ul>

</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.Property.Address, new { autocomplete = "off", placeholder = "Address", Name = "Address", @class = "form-control" })

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

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

</div>

<div class="form-group">

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

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

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

</div>

<div class="form-group">

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

@\*<input type="text" class="form-control" value="Australia" name="Country" id="user-job-Country" />\*@<!-- Dynamic -->

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

</div>

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

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

<span>When</span>

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

<div class="row">

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

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="Date" id="date" name="date" />\*@

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

</div>

</div>

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

<div class="form-group">

@Html.TextBoxFor(m => m.Time, new { autocomplete = "off", placeholder = "Time", Name = "Time", @class = "form-control time-picker1" })

@\*@Html.TextBoxFor(m => m.Time, new { autocomplete = "off", placeholder = "Time", Name = "Time", @class = "form-control time-picker" })\*@

@\*<input id="timepicker\_ampm\_dark" class="timepicker form-control" type="time" placeholder="Time">\*@

@\*<input type="text" class="form-control time-picker" placeholder="Time" name="time" id="time" />\*@

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

</div>

</div>

<div class="input-field col s12">

</div>

</div>

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

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

<span>Price</span>

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

<div class="form-group">

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

@\*<input type="text" class="form-control" placeholder="Price" name="Price" id="user-job-Price" />\*@

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

</div>

</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-job-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-group">

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

@\*<textarea rows="3" class="form-control" id="user-job-Instructions" name="Instructions" placeholder="Instructions"></textarea>\*@

</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.Property.BedroomNum, new { autocomplete = "off", placeholder = "Bedrooms", Name = "Bedrooms", @class = "form-control" })

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

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

</div>

<div class="form-group">

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

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

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

</div>

<div class="form-group">

@Html.TextAreaFor(m => m.Property.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" name="Description" id="user-job-Description"></textarea>\*@

</div>

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

@{

var cleanerList = new EntitiesLibrary.Data.WebService().fetchUserSelectedCleaners(Convert.ToInt64(Data.UserId)).ToList();

}

@if (cleanerList.Count > 0)

{

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

<span>Cleaner (Optional)</span>

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

<div class="form-group">

<select class="form-control" id="slt-Selectedleaner">

<option value="0" data-cleanerid=" 0">Select cleaner</option>

@{

foreach (var item in cleanerList)

{

<option value="@item.CLEANERID" data-cleanerid="@item.CLEANERID" data-phone="@item.PHONE" data-pic="@item.PROFILEIMAGE">@item.FIRSTNAME @item.LASTNAME</option>

}

}

</select>

</div>

<div class="form-group" id="div-selected-cleaner" style="display:none;">

<div class="cleaners">

<img class="img-responsive" id="img-pic" src="">

<div class="cleanerName">

<h4 id="h4-cleanername">Cleaner</h4>

<h5><i class="fa fa-phone"></i>&nbsp;<label id="lbl-phone"> Phone</label> </h5>

</div>

</div>

</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="Submit" id="btn-dummy-job-post" class="btn btn-default view-job-btn view-btn" style="width:100%;" />

<input type="submit" value="Submit" id="btn-job-post" 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 -->

<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\_file1').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);

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

$('#IsNewProperty').val("true");

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

}

reader.readAsDataURL(imageFile);

}

}

}

$(document).on('change', ".my\_file1", function () {

previewPropertyImage(this, $(this));

});

</script>

<script>

$('#slt-Selectedleaner').on('change', function () {

var source = $(this);

var cleanerId = $("#slt-Selectedleaner option:selected").attr('data-cleanerid');

$('#CleanerId').val(cleanerId);

var phone = $("#slt-Selectedleaner option:selected").attr('data-phone');

var pic = $("#slt-Selectedleaner option:selected").attr('data-pic');

var cleanerName = $("#slt-Selectedleaner option:selected").text();

if (cleanerId == 0) {

$('#lbl-phone').text(' Phone');

$('#h4-cleanername').text('Name');

$('#img-pic').attr("src", "/Content/images/icons/profile-icon.png");

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

$('#div-selected-cleaner').hide();

}

else {

$('#lbl-phone').text(' ' + phone);

$('#h4-cleanername').text(cleanerName);

$('#img-pic').attr("src", pic);

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

$('#div-selected-cleaner').show();

}

});

</script>

<script>

function getTimezoneName() {

timezone = jstz.determine();

return timezone.name();

}

$(document).ready(function () {

$('#TimeZone').val(getTimezoneName());

});

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

var source = $(this);

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

});

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

var source = $(this);

$('#IsNewProperty').val("true");

$('#Property\_JobType').val($(source).val());

});

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

var source = $(this);

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

$('#IsNewProperty').val("true");

if (propertyTYpe == '0') {

$("#radio-worktype-Meet\_Greet , #radio-worktype-Cleaning").attr('disabled', false);

}

else {

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

$('#Property\_PropertyType').val(propertyTYpe);

if (propertyTYpe == 'Domestic' || propertyTYpe == 'Commercial') {

$("#radio-worktype-Meet\_Greet").prop("checked", false)

$("#radio-worktype-Meet\_Greet").attr('disabled', true);

$("#radio-worktype-Cleaning").prop("checked", true)

}

else {

$("#radio-worktype-Meet\_Greet").attr('disabled', false);

}

}

});

$(document).on('change', '#slct-user-property', function () {

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

if (propertyId == "0") {

$('#IsNewProperty').val("true");

$('#PropertyId').val("0");

}

else {

$('#IsNewProperty').val("false");

}

//if (propertyId != "0") {

// $('#IsNewProperty').val("false");

//}

if (propertyId != "0") {

$.ajax({

url: '/User/getPropertyByID/' + propertyId,

type: "GET",

dataType: 'json',

success: function (result) {

if (result.Status) {

var json = $.parseJSON(result.Data);

console.log(json);

if (json != null) {

if (json.JobTypeId == parseInt(2)) {

$("#radio-jobtype-Business").prop("checked", true)

$('#Property\_JobType').val("Business");

}

else if (json.JobTypeId == parseInt(1)) {

$("#radio-jobtype-Residential").prop("checked", true)

$('#Property\_JobType').val("Residential");

}

//$("#user-job-selct-propertytype option").each(function () {

// if ($(this).val() == json.PropertyType) {

// $(this).attr('selected', 'selected');

// }

//});

$('#user-job-selct-propertytype').val(json.PropertyType);

$('#Property\_PropertyType').val(json.PropertyType);

if (json.PropertyType == '0') {

$("#radio-worktype-Meet\_Greet , #radio-worktype-Cleaning").attr('disabled', false);

}

else {

if (json.PropertyType == 'Domestic' || json.PropertyType == 'Commercial') {

$("#radio-worktype-Meet\_Greet").prop("checked", false)

$("#radio-worktype-Meet\_Greet").attr('disabled', true);

$("#radio-worktype-Cleaning").prop("checked", true)

$('#WorkType').val('Cleaning');

}

else {

$("#radio-worktype-Meet\_Greet").attr('disabled', false);

}

}

$('#PropertyId').val(json.PropertyId);

$('#Property\_PropertyName').val(json.PropertyName);

$('#Property\_FirstName').val(json.FirstName);

$('#Property\_LastName').val(json.LastName);

$('#Property\_Phone').val(json.Phone);

$('#Property\_Address').val(json.Address);

$('#Property\_Suburb').val(json.Suburb);

$('#Property\_State').val(json.State);

$('#Property\_BedroomNum').val(json.BedroomNumString);

$('#Property\_BathroomNum').val(json.BathroomNumString);

$('#Property\_Description').val(json.Description);

//$("#user-job-selct-property select").val(json.PropertyType);

$("#user-job-selct-property").val(json.PropertyType);

$('#Property\_Latitude').val(json.Latitude);

$('#Property\_Longitude').val(json.Longitude);

var template = "";

$.each(json.PropertyImagesFile, function (i) {

//template = template + '<li><input type="file" class="my\_file" style="display:none;" /><input type="image" style="max-width: 76px;" class="user-property-images"data-src="" src="http://202.191.65.67:4106' + json.PropertyImages[i] + '" /></li>';

template = template + '<li><input type="file" class="my\_file1" style="display:none;" /><input type="image" style="max-width: 76px;" class="user-property-images" src= "' + json.PropertyImagesFile[i].Filepath + '" /> <input type="hidden" class="hfn-base64" data-filepath="' + json.PropertyImagesFile[i].Filepath + '" data-fileid="' + json.PropertyImagesFile[i].Id + '"/></li>';

});

for (var i = 0; i < parseInt(4) - json.PropertyImagesFile.length; i++) {

template = template + '<li> <input type="file" class="my\_file1" 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" data-filepath="" data-fileid=""/></li>';

}

//console.log(template);

$('.user-property-images-ul').html(template);

}

}

else {

toastr.error("An error has occured.pls try again");

}

}

});

}

});

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

var status = true;

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

if (this.name != "Instructions" && this.name != 'prpertyname') {

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();

var fileId = $(currentElement).parent().find('.hfn-base64').attr('data-fileid');

var filepath = $(currentElement).parent().find('.hfn-base64').attr('data-filepath');

var newobj = new Object();

newobj.base64 = value;

newobj.fileId = fileId;

newobj.filepath = filepath;

obj.PropertyImagesList.push(newobj);

});

obj.PropertyName = $('#Property\_PropertyName').val();

obj.FirstName = $('#Property\_FirstName').val();

obj.LastName = $('#Property\_LastName').val();

obj.Phone = $('#Property\_Phone').val();

obj.Address = $('#Property\_Address').val();

obj.Suburb = $('#Property\_Suburb').val();

obj.State = $('#Property\_State').val();

obj.Country = $('#Property\_Country').val();

obj.BedroomNum = $('#Property\_BedroomNum').val();

obj.BathroomNum = $('#Property\_BathroomNum').val();

obj.Description = $('#Property\_Description').val();

obj.IsJob = true;

obj.Latitude = $('#Property\_Latitude').val();

obj.Longitude = $('#Property\_Longitude').val();

obj.JobType = $('#Property\_JobType').val();

//obj.PropertyType = $('#slct-user-property').find(":selected").val();

obj.PropertyType = $('#Property\_PropertyType').val();

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

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

$('#btn-job-post').trigger('click');

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

}

}, 300);

});

$(document).on('keydown', "#Property\_Phone , #Price, #Property\_BedroomNum , #Property\_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();

}

if (this.name == 'Bathrooms' || this.name == 'Bedrooms' || this.name == 'phone') {

$('#IsNewProperty').val("true");

}

});

$(document).on('keydown', "#Property\_PropertyName , #Property\_FirstName, #Property\_LastName , #Property\_BathroomNum , #Property\_Address , #Property\_Suburb ,#Property\_State,#Property\_Description", function (e) {

$('#IsNewProperty').val("true");

});

function postjobSuccess(data) {

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

if (data.Status) {

toastr.success("Job posted");

setTimeout(function () {

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

}, 500);

}

else {

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

toastr.error(data.Message);

}

}

</script>

<style>

.wickedpicker {

max-width: 164px;

max-height: 100px;

}

</style>

<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('Property\_Address'), { componentRestrictions: { country: 'au' } });

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

var state = new google.maps.places.Autocomplete(document.getElementById('Property\_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 = $('#Property\_Suburb').val();

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

$('#Property\_Suburb').val(cityFieldValue);

});

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

var stateFieldValue = $('#Property\_State').val();

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

$('#Property\_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 = $('#Property\_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)

$('#Property\_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)

$('#Property\_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);

}

}

$('#Property\_Address').val(addressFieldValue);

$('#Property\_Latitude').val(latitude);

$('#Property\_Longitude').val(longitude);

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

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

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

});

});

</script>

<script src="~/Content/js/timepicker.js"></script>

<script>

//$('.time-picker').timepicker();

</script>

<link href="~/Content/Plugin/Timepicker/wickedpicker.css" rel="stylesheet" />

<script src="~/Content/Plugin/Timepicker/wickedpicker.js"></script>

<script type="text/javascript">

$('.time-picker1').wickedpicker({

twentyFour: false, showSeconds: false

});

// $('.timepicker-24').wickedpicker({twentyFour: true});

</script>