|  |
| --- |
| **Integer Number Jquery validator** |
| $(document.body).on("keypress", ".number", function (e) {  if (e.which != 8 && e.which != 0 && (e.which < 48 || e.which > 57)) {  returnfalse;  }  }); |
| **Decimal Number Jquery validator** |
| $(document.body).on("keypress", ".decimalnumber", function (evt) {  var charCode = (evt.which) ? evt.which : event.keyCode  if (  (charCode != 46 || $(this).val().indexOf('.') != -1) &&// “.” CHECK DOT, AND ONLY ONE.  (charCode < 48 || charCode > 57))  returnfalse;  }); |
| **Year Jquery validator** |
| $(document.body).on('blur', '.yearvalidate', function () {  var current\_year = new Date().getFullYear();  if ($(this).val() != null && $(this).val().trim() != "") {  if (($(this).val().length == 4) && (Number($(this).val()) <= current\_year)) {  $(this).css('border-color', '');  }  else {  $(this).css('border-color', 'red');  }  }  else {  $(this).css('border-color', '');  }  }); |
| **Jquery required field blur** |
| $(".required").blur(function () {  if ($(this).val() != null&& $(this).val().trim() != "")  {  $(this).css('border-color', '');  }  else  {  $(this).css('border-color', 'red');  }  }); |
| **Thumb nail image view** |
| function checkFileExtension(elem) {  var filePath = elem.value;  if (filePath.indexOf('.') == -1)  return false;  var validExtensions = new Array();  var ext = filePath.substring(filePath.lastIndexOf('.') + 1).toLowerCase();  validExtensions[0] = 'jpg';  validExtensions[1] = 'jpeg';  validExtensions[2] = 'bmp';  validExtensions[3] = 'png';  for (var i = 0; i < validExtensions.length; i++) {  if (ext == validExtensions[i])  return true;  }  alert('The file extension ' + ext.toUpperCase() + ' is not allowed!');  elem.value = null;  return false;  }  function showMyImage(fileInput) {  if (checkFileExtension(fileInput) == true) {  var files = fileInput.files;  for (var i = 0; i < files.length; i++) {  var file = files[i];  var imageType = /image.\*/;  if (!file.type.match(imageType)) {  continue;  }  var img = document.getElementById("thumbnil");  img.file = file;  var reader = new FileReader();  reader.onload = (function (aImg) {  return function (e) {  aImg.src = e.target.result;  };  })(img);  reader.readAsDataURL(file);  }  }  } |
| **Sql Query to get multiple rows data is single coulumn** |
| SELECTAttrId,AttrName,  Options=STUFF  (  (  SELECTDISTINCT', '+CAST(t2.[Title]ASVARCHAR(MAX))  FROM[dbo].[AttrDetails]t2  WHEREt2.AttrId=t1.AttrId  FORXMLPATH('')  ),1,1,''  )  FROM[dbo].[ItemAttribute]t1  GROUPBYAttrId,AttrName |
| **Sql code to get the dates of all Friday,Sunday in a month** |
| DECLARE@startDateDATETIME,@endDateDATETIME,@CurrentDateDATETIME  set@CurrentDate=GETDATE();  set@startDate=DATEADD(MM,DATEDIFF(MM,0,@CurrentDate),0);  set@endDate=DATEADD(MM,DATEDIFF(MM,0,DATEADD(MM,1,@startDate)),-1);  ;WITHCalenderAS (  SELECT@startDateASdt  UNIONALL  SELECTdt+ 1 FROMCalender  WHEREdt+ 1 <=@endDate  )  SELECT  dt  ,NameMonth=DATENAME(Month,dt)  ,NameDay=DATENAME(Weekday,dt)  ,WeekofYr=DATEPART(WEEK,dt)FROMCalender  WHEREDATENAME(Weekday,dt)IN('Friday')  Option(MaxRecursion 0) |
| Jquery to generate multiple html partial parts |
| var room = 1;  function Add\_fields() {  room++;  var objTo = document.getElementById('dynamic\_fields');  var divtest = document.createElement("div");  divtest.setAttribute("class", "add\_rooms removeclass");  //var rdiv = 'removeclass';  $.ajax({  url: '@Url.Action("GetPartialRoom", "Booking")',  type: 'POST',  data: { id: '' },  success: function (data) {  divtest.innerHTML = data;  objTo.appendChild(divtest);  var divcount = 0;  $('#dynamic\_fields .itemheading').each(function () {  divcount = divcount + 1;  $(this).html('Room ' + divcount);  //=========================  //var dvalue = $(this).parents('.row.mt10').attr('data-item');  //$('#dv\_' + dvalue).val(divcount);  calNopersons();  //=========================  //if (divcount != 1) {  // $(this).siblings('.removebtnclass').show();  //}  });  },  error: function (req, status, error) {  }  });  }  function remove\_fields(elem) {  var icount = 0;  $('#dynamic\_fields .itemheading').each(function () {  icount++;  })  if (icount > 1) {  $(elem).parents('#dynamic\_fields .removeclass').remove();  }  //if (icount != 1) {  // $(this).siblings('.removebtnclass').show();  //}  //else if (icount == 1)  //{  // $(this).siblings('.removebtnclass').hide();  //}  var divcount = 0;  $('#dynamic\_fields .itemheading').each(function () {  divcount = divcount + 1;  $(this).html('Room ' + divcount);  //var dvalue = $(this).parents('.row.mt10').attr('data-item');  //$('#dv\_' + dvalue).val(divcount);  });  calNopersons();  } |
| Jquery to create Age dropdown list for each children |
| function createageddl(elem) {  var chv = $(elem).val();  //if (Number(chv) > 0) {  // $('.childAgeCaption').show();  //}  //else {  // $('.childAgeCaption').hide();  //}  var totalitems = '';  var childs = $(elem).parents('.form\_add\_1').find('.childagediv').html();  var x = 0;  $(elem).parents('.form\_add\_1').find('.childagediv .childcls').each(function () {  if (x < Number(chv)) {  x++;  }  else {  $(this).remove();  }  });  var item =  '<div class="col-md-3 childcls">' +  '<label>Age(s)</label>' +  '<select name="ChildAge" class="form-control ChildAge">' +  '<option value="1">1 Year</option>' +  '<option value="2">2 Years</option>' +  '<option value="3">3 Years</option>' +  '<option value="4">4 Years</option>' +  '<option value="5">5 Years</option>' +  '<option value="6">6 Years</option>' +  '<option value="7">7 Years</option>' +  '<option value="8">8 Years</option>' +  '<option value="9">9 Years</option>' +  '<option value="10">10 Years</option>' +  '<option value="11">11 Years</option>' +  '</select>' +  '</div>' +  '</div>';  if (x == 0) {  for (var i = 0; i < Number(chv) ; i++) {  totalitems = totalitems + item;  $(elem).parents('.form\_add\_1').find('.childagediv').html(totalitems);  }  }  else {  for (var i = x; i < Number(chv) ; i++) {  totalitems = totalitems + item;  }  $(elem).parents('.form\_add\_1').find('.childagediv').append(totalitems);  }  calNopersons();  } |
| **Jquery Email Validator** |
| $(".email\_required").blur(function () {  if ($(this).val() != null&& $(this).val().trim() != "") {  var emailReg = new RegExp(/^(("[\w-\s]+")|([\w-]+(?:\.[\w-]+)\*)|("[\w-\s]+")([\w-]+(?:\.[\w-]+)\*))(@@((?:[\w-]+\.)\*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$)|(@@\[?((25[0-5]\.|2[0-4][0-9]\.|1[0-9]{2}\.|[0-9]{1,2}\.))((25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.){2}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\]?$)/i);  var valid = emailReg.test($(this).val());  if (!valid) {  $(this).css('border-color', 'red');  } else {  $(this).css('border-color', '');  }  }  else  {  $(this).css('border-color', 'red');  }  }); |
| **Jquery Form Validator Code** |
| $(function (){  $("#bookingfrm").submit(function (event) {  // events: required, number, decimal, email\_required  var chk\_required = true;  var chk\_email\_required = true;  var chk\_email\_format = true;  $(this).find('.required').each(function () {  if ($(this).val() == null || $(this).val().trim() == "") {  $(this).css('border-color', 'red');  chk\_required = false;  }  });  $(this).find('.email\_required').each(function () {  if ($(this).val() == null || $(this).val().trim() == "") {  $(this).css('border-color', 'red');  chk\_email\_required = false;  }  });  $(this).find('.email\_required').each(function () {  if ($(this).val() != null || $(this).val().trim() != "") {  var emailReg = new RegExp(/^(("[\w-\s]+")|([\w-]+(?:\.[\w-]+)\*)|("[\w-\s]+")([\w-]+(?:\.[\w-]+)\*))(@@((?:[\w-]+\.)\*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)$)|(@@\[?((25[0-5]\.|2[0-4][0-9]\.|1[0-9]{2}\.|[0-9]{1,2}\.))((25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\.){2}(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[0-9]{1,2})\]?$)/i);  var valid = emailReg.test($(this).val());  if (!valid)  {  $(this).css('border-color', 'red');  chk\_email\_format = false;  }  else  {  $(this).css('border-color', '');  }  }  });  if (chk\_required == true&& chk\_email\_required == true&& chk\_email\_format == true) {  returntrue;  }  else//if (chk\_required == false || chk\_email\_required == false || chk\_email\_format == false)  {  event.preventDefault();  }  });  }); |
| **Post a form using Jquery** |
| $('#LoginFrm').submit(function (e) {  e.preventDefault();  $('#loaderdiv').show();  $.ajax({  type: $(this).attr('method'),  url: $(this).attr('action'),  data: $(this).serialize(),  success: function (result) {  var item = "";  if (!result.success) {  //var msgarray = [];  var msgarray = new Array();  msgarray = result.msg.split("#");  var i = 0;  for (i = 0; i < msgarray.length; i++) {  if (msgarray[i] != '') {  item = item + '<p style="color:#ac0106">' + msgarray[i] + '</p>';  }  }  $('#LoginMsg').html(item);  }  else {  //$('#LoginMsg').html('<p style="color:green">' + result.msg + '</p>');  window.location = window.location;  }  $('#loaderdiv').hide();  }  });  }) |
| **Login with facebook** |
| public ActionResult LoginWithFaceBook()  {  FaceBookConnect.Authorize("user\_photos,email", string.Format("{0}://{1}/{2}", Request.Url.Scheme, Request.Url.Authority, "Account/loginwithfacebookaction/"));  return new EmptyResult();  }  public ActionResult loginwithfacebookaction(User\_RegisterByFaceBook model)  {  string fullName = "", fn = "", ln = "";  string table = "Login\_Table";  string Emailid = "User\_Name";  FaceBookConnect.API\_Key = FAPI\_Keyy;  FaceBookConnect.API\_Secret = FAPI\_Secrett;  User\_RegisterByFaceBook faceBookUser = new User\_RegisterByFaceBook();  if (Request.QueryString["error"] == "access\_denied")  {  ViewBag.Message = "User has denied access.";  }  else  {  string code = Request.QueryString["code"];  if (!string.IsNullOrEmpty(code))  {  string data = FaceBookConnect.Fetch(code, "me?fields=id,name,email");  faceBookUser = new JavaScriptSerializer().Deserialize<User\_RegisterByFaceBook>(data);  faceBookUser.PictureUrl = string.Format("https://graph.facebook.com/{0}/picture", faceBookUser.Id);  }  }  ViewBag.FID = (!string.IsNullOrEmpty(faceBookUser.Id)) ? faceBookUser.Id : "N/A";  ViewBag.FName = (!string.IsNullOrEmpty(faceBookUser.Name)) ? faceBookUser.Name : "N/A";  ViewBag.FEmail = (!string.IsNullOrEmpty(faceBookUser.Email)) ? faceBookUser.Email : "N/A";  ViewBag.FProfilePhoto = (!string.IsNullOrEmpty(faceBookUser.PictureUrl)) ? faceBookUser.PictureUrl : "N/A";  if (ViewBag.FName != "N/A")  {  fullName = ViewBag.FName;  var names = fullName.Split(' ');  if (names.Length == 1)  {  fn = names[0];  ln = names[0];  }  else if (names.Length == 2)  {  fn = names[0];  ln = names[1];  }  else if (names.Length == 3)  {  fn = names[0];  ln = names[2];  }  }  ViewBag.UserFirstName = (!string.IsNullOrEmpty(fn)) ? fn : "N/A";  ViewBag.UserLastName = (!string.IsNullOrEmpty(ln)) ? ln : "N/A";  if (ViewBag.FID != "N/A")  {  model.Register\_Id = ViewBag.FID;  }  else  {  model.Register\_Id = "";  }  if (ViewBag.FEmail != "N/A")  {  model.EmailorUsername = ViewBag.FEmail;  }  else  {  model.EmailorUsername = "";  }  if (ViewBag.UserFirstName != "N/A")  {  model.First\_Name = ViewBag.UserFirstName;  }  else  {  model.First\_Name = "";  }  if (ViewBag.UserLastName != "N/A")  {  model.Last\_Name = ViewBag.UserLastName;  }  else  {  model.Last\_Name = "";  }  if ((!string.IsNullOrEmpty(model.Register\_Id)) && (!string.IsNullOrEmpty(model.First\_Name)) && (!string.IsNullOrEmpty(model.Last\_Name)) && (!string.IsNullOrEmpty(model.EmailorUsername)))  {  if (!dl.checkExists(table, Emailid, model.EmailorUsername))  {  int i = dl.RegisterUsingFacebook(model);  if (i > 0)  {  string EntryBy = "";  DataSet dsfetchh = dl.Inline\_Process("select EntryBy,Register\_Id from [dbo].[Login\_Table] where Register\_Id='" + model.Register\_Id + "'");  if (dsfetchh != null && dsfetchh.Tables[0].Rows.Count > 0)  {  EntryBy = dsfetchh.Tables[0].Rows[0]["EntryBy"].ToString();  }  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = model.Register\_Id;  loginCookieNHST["FName"] = model.First\_Name;  loginCookieNHST["LName"] = model.Last\_Name;  loginCookieNHST["EntryBy"] = EntryBy;  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["RegSuccess"] = "Register successfully!";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Oops!Something is going wrong.";  }  }  else  {  DataSet ds = dl.LoginWithFacebook(model.EmailorUsername);  if (ds.Tables[0].Rows.Count > 0)  {  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = ds.Tables[0].Rows[0]["Register\_Id"].ToString();  loginCookieNHST["FName"] = ds.Tables[0].Rows[0]["First\_Name"].ToString();  loginCookieNHST["LName"] = ds.Tables[0].Rows[0]["Last\_Name"].ToString();  loginCookieNHST["EntryBy"] = ds.Tables[0].Rows[0]["EntryBy"].ToString();  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["LogSuccess"] = "Login successfully done.";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Invalid Username or Password...try again!";  return RedirectToAction("Login", "Account");  }  }  }  else  {  TempData["LogError"] = "Failed to login with facebook.";  return RedirectToAction("Login", "Account");  }  return RedirectToAction("Login", "Account");  } |
| **Login with Google** |
| public ActionResult GoogleLogin()  {  var Googleurl = "<https://accounts.google.com/o/oauth2/auth?response_type=code&redirect_uri=>" + googleplus\_redirect\_url + "&scope=https://www.googleapis.com/auth/userinfo.email%20https://www.googleapis.com/auth/userinfo.profile&client\_id=" + googleplus\_client\_id;  Session["loginWith"] = "google";  return Redirect(Googleurl);  //return View();  }  public ActionResult LoginToGoogle(string code = "")  {  string table = "Login\_Table";  string Emailid = "User\_Name";  try  {  var url = Request.Url.Query;  if (url != "")  {  string queryString = url.ToString();  char[] delimiterChars = { '=' };  string[] words = queryString.Split(delimiterChars);  //string code = words[1];  if (code != null)  {  //get the access token  HttpWebRequest webRequest = (HttpWebRequest)WebRequest.Create("https://accounts.google.com/o/oauth2/token");  webRequest.Method = "POST";  Parameters = "code=" + code + "&client\_id=" + googleplus\_client\_id + "&client\_secret=" + googleplus\_client\_sceret + "&redirect\_uri=" + googleplus\_redirect\_url + "&grant\_type=authorization\_code";  byte[] byteArray = Encoding.UTF8.GetBytes(Parameters);  webRequest.ContentType = "application/x-www-form-urlencoded";  webRequest.ContentLength = byteArray.Length;  Stream postStream = webRequest.GetRequestStream();  // Add the post data to the web request  postStream.Write(byteArray, 0, byteArray.Length);  postStream.Close();  WebResponse response = webRequest.GetResponse();  postStream = response.GetResponseStream();  StreamReader reader = new StreamReader(postStream);  string responseFromServer = reader.ReadToEnd();  GooglePlusAccessToken serStatus = JsonConvert.DeserializeObject<GooglePlusAccessToken>(responseFromServer);  if (serStatus != null)  {  string accessToken = string.Empty;  accessToken = serStatus.access\_token;  if (!string.IsNullOrEmpty(accessToken))  {  // This is where you want to add the code if login is successful.  //getgoogleplususerdataSer(accessToken);  WebClient client = new WebClient();  var urlProfile = "<https://www.googleapis.com/oauth2/v1/userinfo?access_token=>" + accessToken;  string downloadString = client.DownloadString(urlProfile);  GoogleUserOutputData \_obj = JsonConvert.DeserializeObject<GoogleUserOutputData>(downloadString);  if (\_obj != null)  {  User\_RegisterByGoogle model = new User\_RegisterByGoogle();  model.googleId = \_obj.id;  string fname = "", lname = "";  if (\_obj.name != null)  {  string[] data = \_obj.name.Split(' ');  if (data.Length == 1)  {  fname = data[0];  lname = data[0];  }  else if (data.Length == 2)  {  fname = data[0];  lname = data[1];  }  else if (data.Length == 3)  {  fname = data[0];  lname = data[2];  }  }  string cc = "", vv = "", propertyid = "";  model.First\_Name = fname;  model.Last\_Name = lname;  model.EmailorUsername = \_obj.email;  model.Register\_Id = model.googleId;  if ((!string.IsNullOrEmpty(model.Register\_Id)) && (!string.IsNullOrEmpty(model.First\_Name)) && (!string.IsNullOrEmpty(model.Last\_Name)) && (!string.IsNullOrEmpty(model.EmailorUsername)))  {  if (!dl.checkExists(table, Emailid, model.EmailorUsername))  {  int i = dl.RegisterUsingGoogle(model);  if (i > 0)  {  string EntryBy = "";  DataSet dsfetchh = dl.Inline\_Process("select EntryBy,Register\_Id from [dbo].[Login\_Table] where Register\_Id='" + model.Register\_Id + "'");  if (dsfetchh != null && dsfetchh.Tables[0].Rows.Count > 0)  {  EntryBy = dsfetchh.Tables[0].Rows[0]["EntryBy"].ToString();  }  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = model.Register\_Id;  loginCookieNHST["FName"] = model.First\_Name;  loginCookieNHST["LName"] = model.Last\_Name;  loginCookieNHST["EntryBy"] = EntryBy;  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["RegSuccess"] = "Register successfully!";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Oops!Something is going wrong.";  }  }  else  {  DataSet ds = dl.LoginWithGoogle(model.EmailorUsername);  if (ds.Tables[0].Rows.Count > 0)  {  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = ds.Tables[0].Rows[0]["Register\_Id"].ToString();  loginCookieNHST["FName"] = ds.Tables[0].Rows[0]["First\_Name"].ToString();  loginCookieNHST["LName"] = ds.Tables[0].Rows[0]["Last\_Name"].ToString();  loginCookieNHST["EntryBy"] = ds.Tables[0].Rows[0]["EntryBy"].ToString();  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["LogSuccess"] = "Login successfully done.";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Invalid Username or Password...try again!";  return RedirectToAction("Login", "Account");  }  }  }  else  {  TempData["LogError"] = "Failed to login with facebook.";  return RedirectToAction("Login", "Account");  }  }  }  }  }  }  }  catch (Exception ex)  {  //throw new Exception(ex.Message, ex);  //Response.Redirect("index.aspx");  }  return RedirectToAction("Login", "Account");  }  public class GooglePlusAccessToken  {  public string access\_token { get; set; }  public string token\_type { get; set; }  public int expires\_in { get; set; }  public string id\_token { get; set; }  public string refresh\_token { get; set; }  }  private async void getgoogleplususerdataSer(string access\_token)  {  try  {  HttpClient client = new HttpClient();  var urlProfile = "<https://www.googleapis.com/oauth2/v1/userinfo?access_token=>" + access\_token;  client.CancelPendingRequests();  HttpResponseMessage output = await client.GetAsync(urlProfile);  if (output.IsSuccessStatusCode)  {  string outputData = await output.Content.ReadAsStringAsync();  GoogleUserOutputData serStatus = JsonConvert.DeserializeObject<GoogleUserOutputData>(outputData);  if (serStatus != null)  {  // You will get the user information here.  }  }  }  catch (Exception ex)  {  //catching the exception  }  }  public class GoogleUserOutputData  {  public string id { get; set; }  public string name { get; set; }  public string given\_name { get; set; }  public string email { get; set; }  public string picture { get; set; }  } |
| **Login with twitter** |
| public ActionResult LoginWithTwitter()  {  if (!TwitterConnect.IsAuthorized)  {  TwitterConnect twitter = new TwitterConnect();  twitter.Authorize(string.Format("{0}://{1}/{2}", Request.Url.Scheme, Request.Url.Authority, "Account/loginwithtwitteraction/"));  }  return new EmptyResult();  }  public ActionResult loginwithtwitteraction(User\_RegisterByTwitter model1)  {  string fullName = "", fn = "", ln = "";  string table = "Login\_Table";  string Emailid = "User\_Name";  TwitterConnect.API\_Key = TAPI\_Keyy;  TwitterConnect.API\_Secret = TAPI\_Secrett;  User\_RegisterByTwitter model = new User\_RegisterByTwitter();  if (TwitterConnect.IsAuthorized)  {  TwitterConnect twitter = new TwitterConnect();  DataTable dt = twitter.FetchProfile();  model.ProfileImageUrl = dt.Rows[0]["profile\_image\_url"].ToString();  model.Name = dt.Rows[0]["name"].ToString();  model.Id = dt.Rows[0]["Id"].ToString();  model.ScreenName = dt.Rows[0]["screen\_name"].ToString();  model.Description = dt.Rows[0]["description"].ToString();  model.StatusesCount = dt.Rows[0]["statuses\_count"].ToString();  model.FollowersCount = dt.Rows[0]["followers\_count"].ToString();  model.FriendsCount = dt.Rows[0]["friends\_count"].ToString();  model.FavouritesCount = dt.Rows[0]["favourites\_count"].ToString();  model.Location = dt.Rows[0]["location"].ToString();  fullName = model.Name;  var names = fullName.Split(' ');  if (names.Length == 1)  {  fn = names[0];  ln = names[0];  }  else if (names.Length == 2)  {  fn = names[0];  ln = names[1];  }  else if (names.Length == 3)  {  fn = names[0];  ln = names[2];  }  model.Register\_Id = model.Id;  model.First\_Name = fn;  model.Last\_Name = ln;  model.EmailorUsername = model.ScreenName;  if ((!string.IsNullOrEmpty(model.Register\_Id)) && (!string.IsNullOrEmpty(model.First\_Name)) && (!string.IsNullOrEmpty(model.Last\_Name)) && (!string.IsNullOrEmpty(model.EmailorUsername)))  {  if (!dl.checkExists(table, Emailid, model.EmailorUsername))  {  int i = dl.RegisterUsingTwitter(model);  if (i > 0)  {  string EntryBy = "";  DataSet dsfetchh = dl.Inline\_Process("select EntryBy,Register\_Id from [dbo].[Login\_Table] where Register\_Id='" + model.Register\_Id + "'");  if (dsfetchh != null && dsfetchh.Tables[0].Rows.Count > 0)  {  EntryBy = dsfetchh.Tables[0].Rows[0]["EntryBy"].ToString();  }  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = model.Register\_Id;  loginCookieNHST["FName"] = model.First\_Name;  loginCookieNHST["LName"] = model.Last\_Name;  loginCookieNHST["EntryBy"] = EntryBy;  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["RegSuccess"] = "Register successfully!";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Oops!Something is going wrong.";  }  }  else  {  DataSet ds = dl.LoginWithTwitter(model.EmailorUsername);  if (ds.Tables[0].Rows.Count > 0)  {  HttpCookie loginCookieNHST = Request.Cookies["loginCookieNHST"];  loginCookieNHST = new HttpCookie("loginCookieNHST");  loginCookieNHST["RegisterId"] = ds.Tables[0].Rows[0]["Register\_Id"].ToString();  loginCookieNHST["FName"] = ds.Tables[0].Rows[0]["First\_Name"].ToString();  loginCookieNHST["LName"] = ds.Tables[0].Rows[0]["Last\_Name"].ToString();  loginCookieNHST["EntryBy"] = ds.Tables[0].Rows[0]["EntryBy"].ToString();  loginCookieNHST.Expires = DateTime.Now.AddHours(1);  Response.Cookies.Add(loginCookieNHST);  TempData["LogSuccess"] = "Login successfully done.";  return RedirectToAction("Dashboard", "Account");  }  else  {  TempData["LogError"] = "Invalid Username or Password...try again!";  return RedirectToAction("Login", "Account");  }  }  }  else  {  TempData["LogError"] = "Failed to login with facebook.";  return RedirectToAction("Login", "Account");  }  }  if (TwitterConnect.IsDenied)  {  ViewBag.Message = "User has denied access.";  }  return RedirectToAction("Login", "Account");  } |
| **Google Translator** |
| <scripttype="text/javascript">  function googleTranslateElementInit() {  new google.translate.TranslateElement({ pageLanguage: 'en', includedLanguages: 'vi', layout: google.translate.TranslateElement.InlineLayout.SIMPLE }, 'google\_translate\_element');  }  </script>  <scripttype="text/javascript"src="//translate.google.com/translate\_a/element.js?cb=googleTranslateElementInit"></script> |
| **Google Map Integration** |
| <scripttype="text/javascript"src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAqb3fT3SbMSDMggMEK7fJOIkvamccLrjA&amp;sensor=false"></script>  <!-- google maps -->  <scripttype="text/javascript">  var locations = [  @{  int loccount = 1;  foreach (DataRow dr in dt.Rows)  {  if (dr["Latitude"].ToString() != ""&& dr["Longitude"].ToString() != "")  {  decimal \_price = 0;  decimal.TryParse(dr["Price"].ToString(), out \_price);  decimal \_area = 0;  decimal.TryParse(dr["Area"].ToString(), out \_area);  @Html.Raw("[\'<div onclick=\"showpropertypopup(this)\" data-item=\""+dr["PropertyId"].ToString()+"\" style=\"cursor:pointer\" class=\"gm-style-iw\" >" +  "<div style=\"display: inline-block; overflow: auto; max-height: 559px; max-width: 654px;\">" +  "<div style=\"overflow: auto;\">" +  "<table cellpadding=\"5\" cellspacing=\"0\" border=\"0\" width=\"100%\" height=\"auto\">" +  "<tbody>" +  "<tr>" +  "<td valign=\"top\" style=\"padding:3px\">" +  "<a onclick=\"" + "Getpropdtls(\"" + dr["PropertyId"].ToString() + "\")" + "\">" +  "<img src=\"" + (dr["FrontImage"].ToString() != "" ? Url.Content("~/Images/Property/" + dr["FrontImage"].ToString()) : Url.Content("~/Content/themes/Sites/images/property-img1.jpg")).ToString() + "\" style=\"max-height:50px;width:30x\">" +  "</a>" +  "</td>" +  "<td rowspan=\"2\" align=\"left\" valign=\"top\" style=\"padding:3px\">" +  "<a onclick=\"" + "Getpropdtls(\"" + dr["PropertyId"].ToString() + "\")" + "\" style=\"color: #464646;text-decoration: none;\">" +  "<b>" + dr["Address"].ToString() + "," + dr["PostCode"].ToString() + "</b>" +  "<br>" +  dr["Beds"].ToString() + " Beds," + dr["Baths"].ToString() + " Baths"  + "<br>" + \_price.ToString("C0") +  "<br>" +  \_area.ToString("#,##0.##") + " " + dr["AreaUnit"].ToString() +  "</a>" +  "</td>" +  "</tr>" +  "</tbody>" +  "</table>" +  "</div>" +  "</div>" +  "</div>\'," + dr["Latitude"].ToString() + ", " + dr["Longitude"].ToString() + ", " + loccount.ToString() + ",'" + dr["PropertyId"].ToString().ToString() + "'],")  }  loccount++;  }  }  //['<h4>Bondi Beach</h4>', -33.890542, 151.274856, 4],  //['<h4>Coogee Beach</h4>', -33.923036, 151.259052, 5],  //['<h4>Cronulla Beach</h4>', -34.028249, 151.157507, 3],  //['<h4>Manly Beach</h4>', -33.80010128657071, 151.28747820854187, 2],  //['<h4>Maroubra Beach</h4>', -33.950198, 151.259302, 1]  ];  var bounds = new google.maps.LatLngBounds();  var gmarkers = [];  var map = new google.maps.Map(document.getElementById('map-canvas'), {  zoom: 5,  //center: new google.maps.LatLng(-33.92, 151.25),  mapTypeId: google.maps.MapTypeId.ROADMAP  });  //map.setTilt(45);  var infowindow = new google.maps.InfoWindow();  var marker, i;  for (i = 0; i < locations.length; i++) {  var position = new google.maps.LatLng(locations[i][1], locations[i][2]);  bounds.extend(position);  marker = new google.maps.Marker({  position: position,  map: map,  url: 'https://developers.google.com/maps/documentation/javascript/images/circle.png',  anchor: new google.maps.Point(10, 10),  scaledSize: new google.maps.Size(10, 17)  });  google.maps.event.addListener(marker, 'click', (function (marker, i) {  returnfunction () {  //infowindow.setContent(locations[i][0]);  //infowindow.open(map, marker);  @\* window.location = '@Url.Action("Details", "Property")?id=' + locations[i][4]\*@  Getpropdtls(locations[i][4]);  }  })(marker, i));  google.maps.event.addListener(marker, 'mouseover', (function (marker, i) {  returnfunction () {  infowindow.setContent(locations[i][0]);  infowindow.open(map, marker);  }  })(marker, i));  //gmarkers = locations.concat();  gmarkers.push(marker);  map.fitBounds(bounds);  }  var boundsListener = google.maps.event.addListener((map), 'bounds\_changed', function (event) {  this.setZoom(5);  google.maps.event.removeListener(boundsListener);  });  function show(i) {  google.maps.event.trigger(gmarkers[i], 'mouseover');  }  </script>  <scripttype="text/javascript"> |
| **Checkbox by name is checked in Jquery** |
| var churchid = $('input[name="selchurh"]:checkbox:checked'); |
| **Jquery to stop copy past** |
| $(document.body).on("keydown", ".decimalnumber", function (event) {  if (event.ctrlKey == true && (event.which == '118' || event.which == '86')) {  event.preventDefault();  }  }); |
| **Edit for Multiple Select List** |
| Function  void fillwithProductCategory(List<string> selected)  {  List<SelectListItem> catlist = new List<SelectListItem>();  DataSet ds = dl.Inline\_Process("Select \* from [dbo].[ProductCategories] where IsActive=1 order by ProductCategory ASC");  if (ds.Tables[0].Rows.Count > 0)  {  for (int i = 0; i < ds.Tables[0].Rows.Count; i++)  {  SelectListItem sl = new SelectListItem();  sl.Text = ds.Tables[0].Rows[i]["ProductCategory"].ToString();  sl.Value = ds.Tables[0].Rows[i]["PCatId"].ToString();  catlist.Add(sl);  }  }  if (selected.Count > 0)  {  SelectList sl = new SelectList(catlist, "Value", "Text");  ViewBag.ProductCategoryList = new MultiSelectList(sl, "Value", "Text", selected);  }  else  {  SelectList sl = new SelectList(catlist, "Value", "Text");  ViewBag.ProductCategoryList = new MultiSelectList(sl, "Value", "Text");  }  }  Controller  public ActionResult editjobopening(string id)  {  List<string> selected = new List<string>();  DataSet ds = dl.Inline\_Process("Select \* from [dbo].[Job\_Product\_Category] where JobId='" + id + "'");    if (ds != null && ds.Tables.Count > 0 && ds.Tables[2].Rows.Count > 0)  {  foreach (DataRow dr in ds.Tables[2].Rows)  {  selected.Add(dr["PCatId"].ToString());  }  }  fillwithProductCategory(selected);  return View();  } |