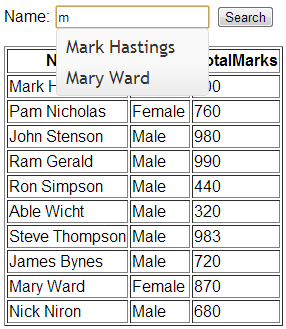
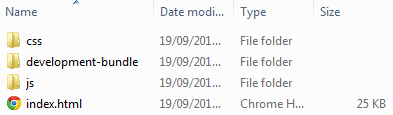
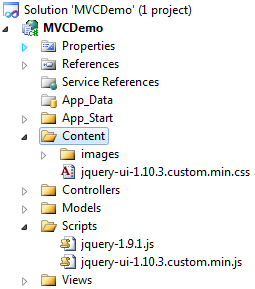
In this video, we will discuss **implementing auto-complete functionality** in an asp.net mvc application using jQuery Autocomplete Widget  
   
  
   
  
**Step 1:** We will be using **tblStudents**table in this demo. Please find the sql script below, to create and populate this table with some data.  
Create Table tblStudents  
(  
 ID int identity primary key,  
 Name nvarchar(50),  
 Gender nvarchar(20),  
 TotalMarks int  
)  
  
Insert into tblStudents values('Mark Hastings','Male',900)  
Insert into tblStudents values('Pam Nicholas','Female',760)  
Insert into tblStudents values('John Stenson','Male',980)  
Insert into tblStudents values('Ram Gerald','Male',990)  
Insert into tblStudents values('Ron Simpson','Male',440)  
Insert into tblStudents values('Able Wicht','Male',320)  
Insert into tblStudents values('Steve Thompson','Male',983)  
Insert into tblStudents values('James Bynes','Male',720)  
Insert into tblStudents values('Mary Ward','Female',870)  
Insert into tblStudents values('Nick Niron','Male',680)

**Step 2:** Create an ado.net entity data model using table **tblStudents.**Change the name of the generated entity from **tblStudent**to **Student**.  
  
**Step 3:**Download autocomplete widget from <http://jqueryui.com/download>. The following folders and files will be downloded.  
   
  
**Step 4:** Open **"js"**folder copy **"jquery-1.9.1.js"** and **"jquery-ui-1.10.3.custom.min.js"**files and paste them in the **"Scripts"**folder of you mvc project. Now open **"css"**folder. This folder will be present in **"ui-lightness"**folder. Copy **"images"**folder and **"jquery-ui-1.10.3.custom.min.css"**file and paste them in **"Content"**folder in your mvc project. If you are following along, at this point your solution explorer should look as shown below.  
   
  
**Step 5:** Right click on the **"Controllers"**folder and add **"Home"**controller. Copy and paste the following code. Please make sure to include **"MVCDemo.Models"**namespace.  
public class HomeController : Controller  
{  
    public ActionResult Index()  
    {  
        DatabaseContext db = new DatabaseContext();  
        return View(db.Students);  
    }  
  
    [HttpPost]  
    public ActionResult Index(string searchTerm)  
    {  
        DatabaseContext db = new DatabaseContext();  
        List<Student> students;  
        if (string.IsNullOrEmpty(searchTerm))  
        {  
            students = db.Students.ToList();  
        }  
        else  
        {  
            students = db.Students  
                .Where(s => s.Name.StartsWith(searchTerm)).ToList();  
        }  
        return View(students);  
    }  
  
    public JsonResult GetStudents(string term)  
    {  
        DatabaseContext db = new DatabaseContext();  
        List<string> students = db.Students.Where(s => s.Name.StartsWith(term))  
            .Select(x => x.Name).ToList();  
        return Json(students, JsonRequestBehavior.AllowGet);  
    }  
}  
  
**Step 6:** Right click on the **"Index"**action method in the **"HomeController"**and add **"Index"**view. Copy and paste the following code.  
@model IEnumerable<MVCDemo.Models.Student>  
  
<link href="~/Content/jquery-ui-1.10.3.custom.min.css" rel="stylesheet" type="text/css" />  
<script src="~/Scripts/jquery-1.9.1.js" type="text/javascript"></script>

<script src="~/Scripts/jquery-ui-1.10.3.custom.min.js" type="text/javascript"></script>  
  
<script type="text/javascript">  
    $(function () {  
        $("#txtSearch").autocomplete({  
            source: '@Url.Action("GetStudents")',  
            minLength: 1  
        });  
    });  
</script>  
  
<div style="font-family:Arial">  
  
@using (@Html.BeginForm())  
{  
    <b>Name: </b>  
    @Html.TextBox("searchTerm", null, new { id = "txtSearch" })  
    <input type="submit" value="Search" />  
}  
  
<table border="1">  
    <tr>  
        <th>  
            @Html.DisplayNameFor(model => model.Name)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.Gender)  
        </th>  
        <th>  
            @Html.DisplayNameFor(model => model.TotalMarks)  
        </th>  
    </tr>  
  
@foreach (var item in Model)  
{  
    <tr>  
        <td>  
            @Html.DisplayFor(modelItem => item.Name)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.Gender)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => item.TotalMarks)  
        </td>  
    </tr>  
}  
</table>  
</div>