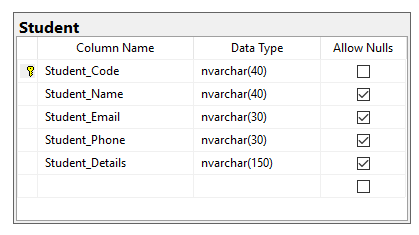
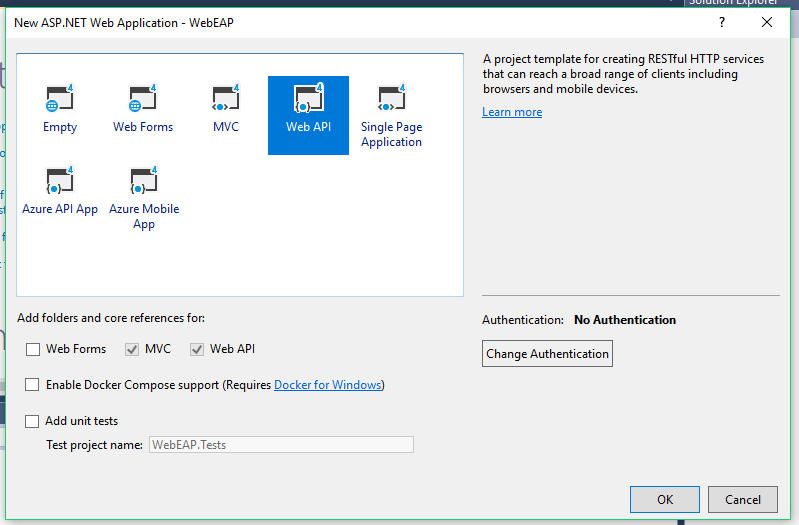
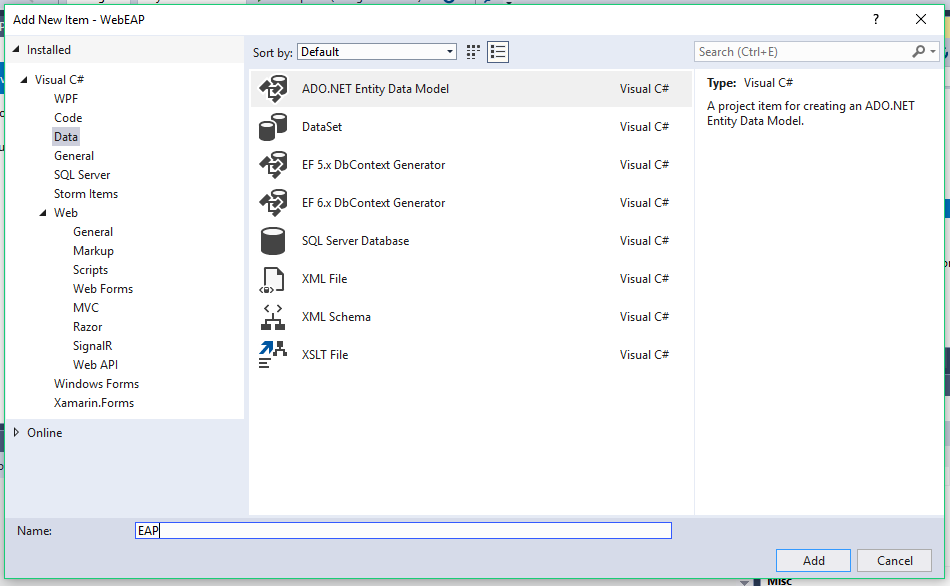
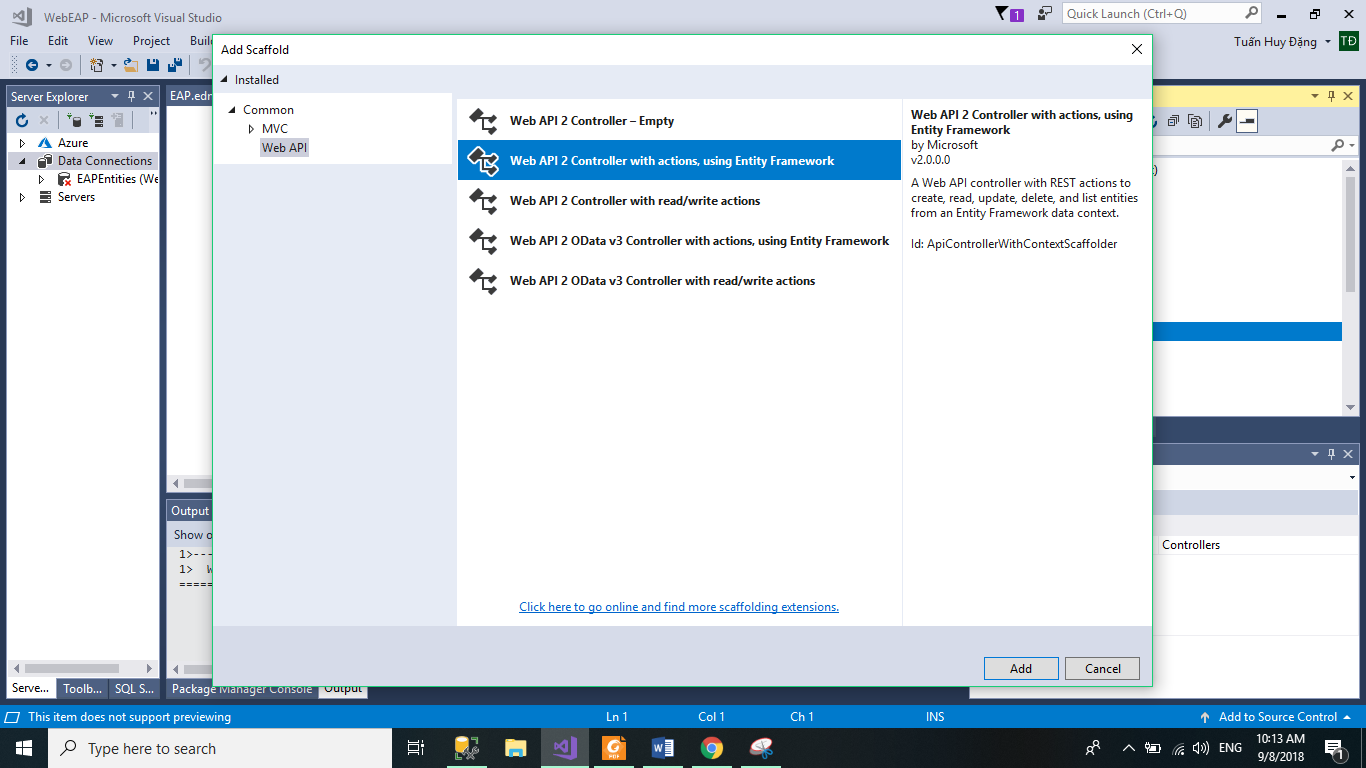
**Reviews EAPs**

1. **CSDL (EAP) – tables Student**
2. **Tạo một Project API-Web**
3. **Sử dụng EntityFramwwork tạo Models**



1. **Tạo controller**



1. **Hiệu chỉnh**

**Trong Application\_Start() bổ sung code convert xml->json**

using System.Web.Http;

using System.Web.Mvc;

using System.Web.Optimization;

using System.Web.Routing;

namespace WebEAP

{

public class WebApiApplication : System.Web.HttpApplication

{

protected void Application\_Start()

{

GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling = Newtonsoft.Json.ReferenceLoopHandling.Ignore;

GlobalConfiguration.Configuration.Formatters.Remove(GlobalConfiguration.Configuration.Formatters.XmlFormatter);

AreaRegistration.RegisterAllAreas();

GlobalConfiguration.Configure(WebApiConfig.Register);

FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);

RouteConfig.RegisterRoutes(RouteTable.Routes);

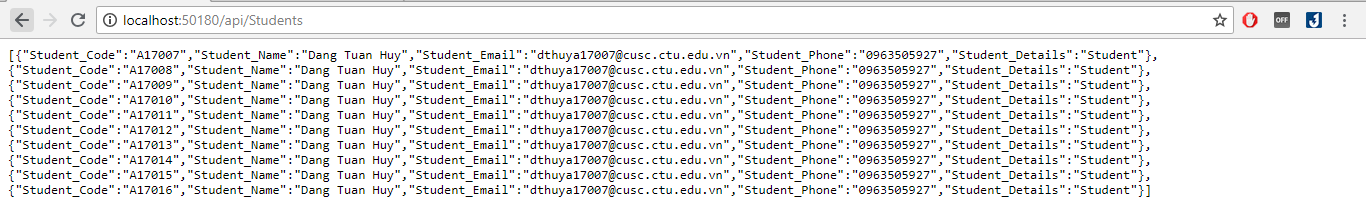
BundleConfig.RegisterBundles(BundleTable.Bundles);

}

}

}

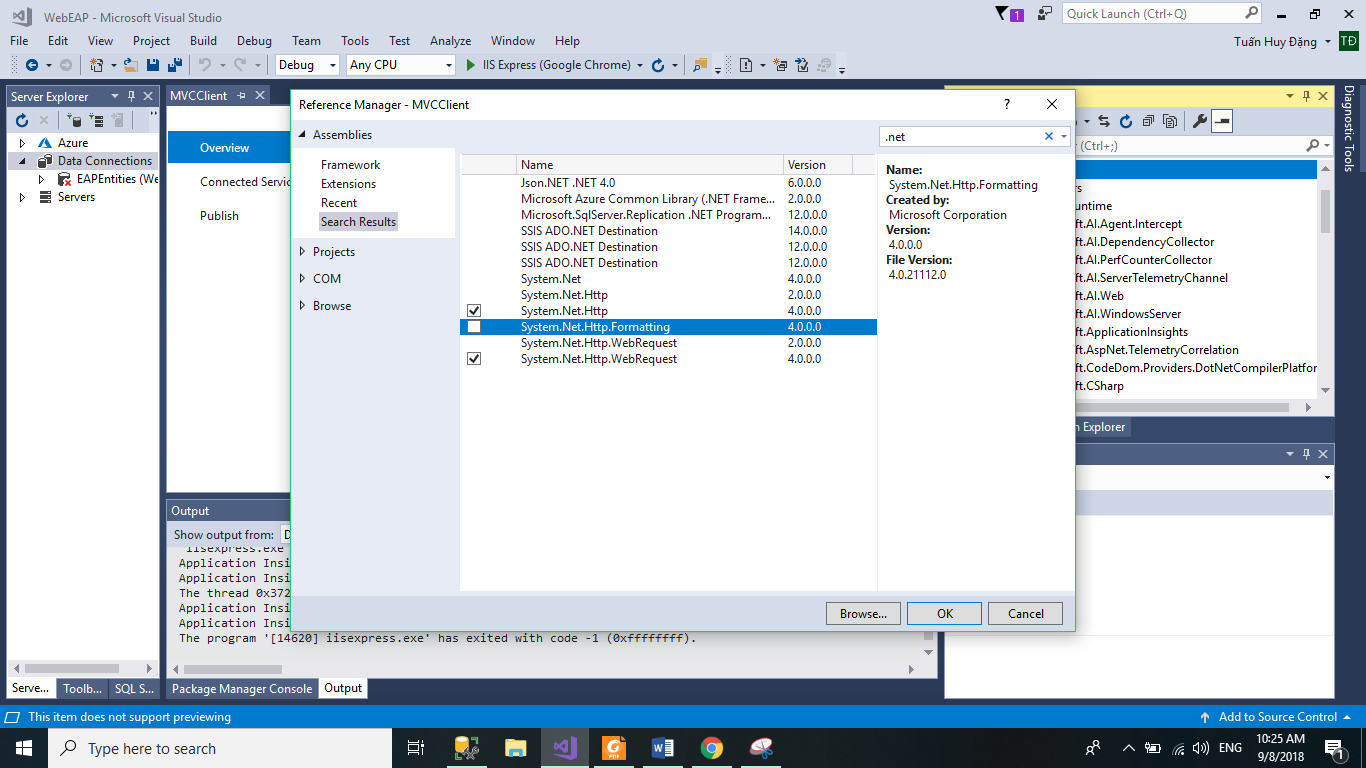
**Kết quả**



1. **API – Client**
2. **Web MVC – Client**

***Trong cùng Solution tạo một Project MVCClient***

***Trong References của MVCClient hiệu chỉnh***

1. **Trong model tạo một Class có tên là Student.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace MVCClient.Models

{

public class Student

{

public string Student\_Code { get; set; }

public string Student\_Name { get; set; }

public string Student\_Email { get; set; }

public string Student\_Phone { get; set; }

public string Student\_Details { get; set; }

}

}

1. **Tạo một controller emty có tên là StudentsController.cs Bổ sung code gọi API**

using MVCClient.Models;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Net.Http.Headers;

using System.Threading.Tasks;

using System.Web;

using System.Web.Mvc;

namespace MVCClient.Controllers

{

public class StudentsController : Controller

{

HttpClient client;

string url = "http://localhost:50180/api/Students";

public StudentsController()

{

client = new HttpClient();

client.BaseAddress = new Uri(url);

client.DefaultRequestHeaders.Accept.Clear();

client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));

}

// Get Student

public async Task<ActionResult> Index()

{

HttpResponseMessage responseMessage = await client.GetAsync(url);

if(responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var students = JsonConvert.DeserializeObject<List<Student>>(responseData);

return View(students);

}

return View("Error");

}

//Form Create

public ActionResult Create()

{

return View(new Student());

}

[HttpPost]

public async Task<ActionResult> Create(Student student)

{

HttpResponseMessage responseMessage = await client.PostAsJsonAsync(url,student);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

//Form Edit

public async Task<ActionResult> Edit(string id)

{

HttpResponseMessage responseMessage = await client.GetAsync(url + "/" + id);

if (responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var student = JsonConvert.DeserializeObject<Student>(responseData);

return View(student);

}

return View("Error");

}

[HttpPost]

public async Task<ActionResult> Edit(string id, Student student)

{

HttpResponseMessage responseMessage = await client.PutAsJsonAsync(url + "/" + id, student);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

//Form Delete

public async Task<ActionResult> Delete(string id)

{

HttpResponseMessage responseMessage = await client.GetAsync(url + "/" + id);

if (responseMessage.IsSuccessStatusCode)

{

var responseData = responseMessage.Content.ReadAsStringAsync().Result;

var student = JsonConvert.DeserializeObject<Student>(responseData);

return View(student);

}

return View("Error");

}

[HttpPost]

public async Task<ActionResult> Delete(string id, Student student)

{

HttpResponseMessage responseMessage = await client.DeleteAsync(url + "/" +id);

if (responseMessage.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return RedirectToAction("Error");

}

}

}

1. **Tạo Views từ model Student**

**Code Index**

@model IEnumerable<MVCClient.Models.Student>

@{

ViewBag.Title = "Index";

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

}

<h2>List Student</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table table-bordered">

<tr>

<th>

@Html.DisplayNameFor(model => model.Student\_Code)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Name)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Email)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Phone)

</th>

<th>

@Html.DisplayNameFor(model => model.Student\_Details)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.Student\_Code)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Name)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Email)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Phone)

</td>

<td>

@Html.DisplayFor(modelItem => item.Student\_Details)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.Student\_Code}) |

@Html.ActionLink("Delete", "Delete", new {id = item.Student\_Code })

</td>

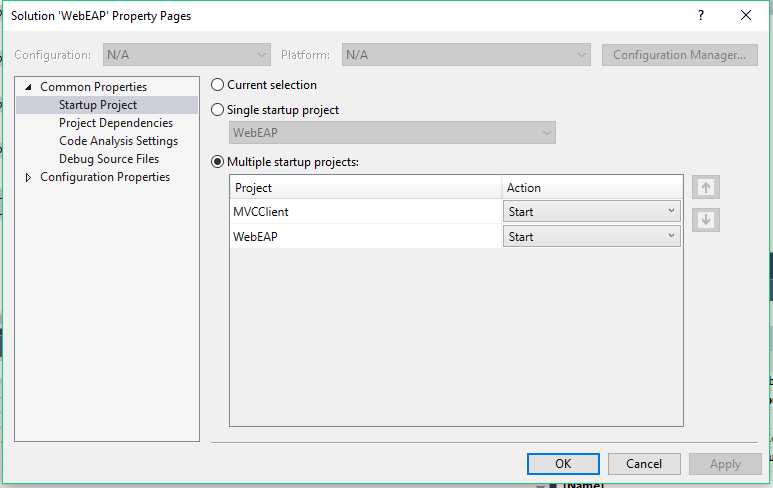
</tr>

}

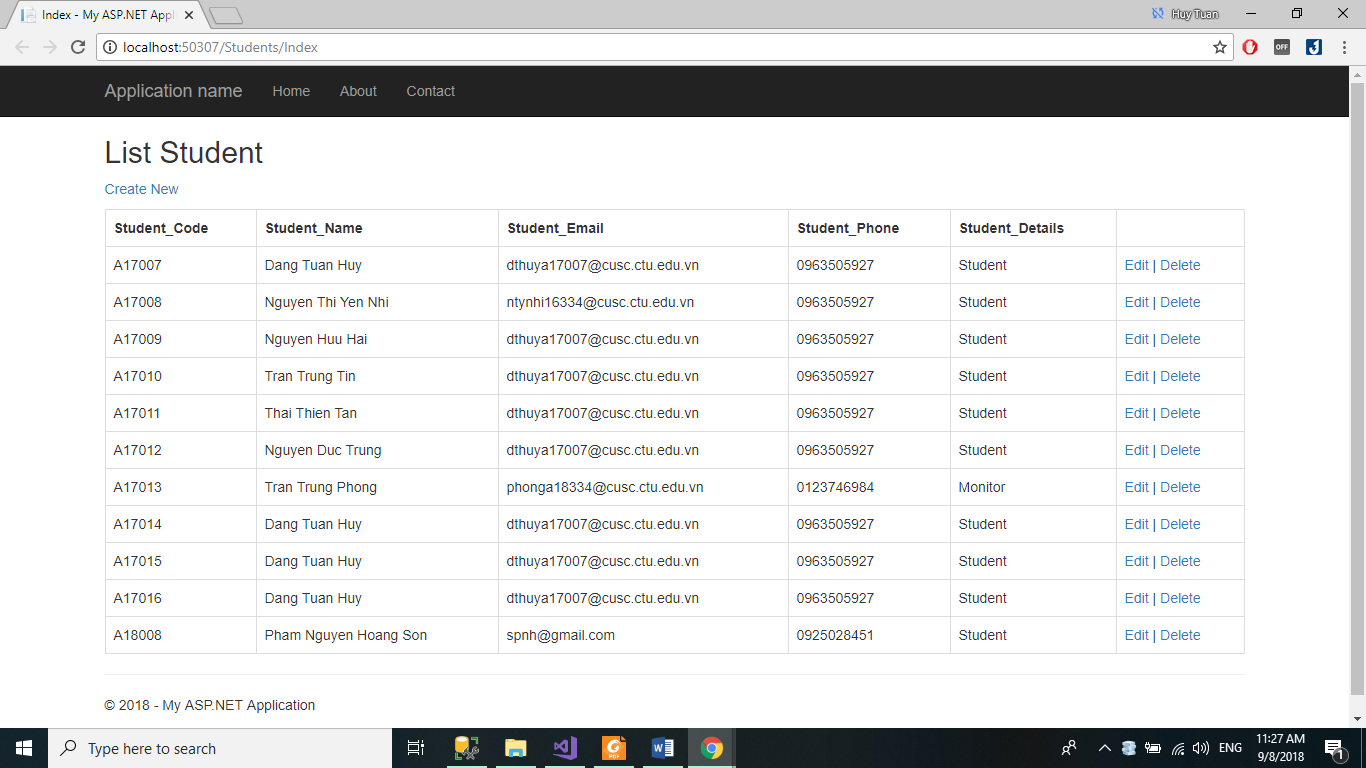
</table>

1. **Kết quả**

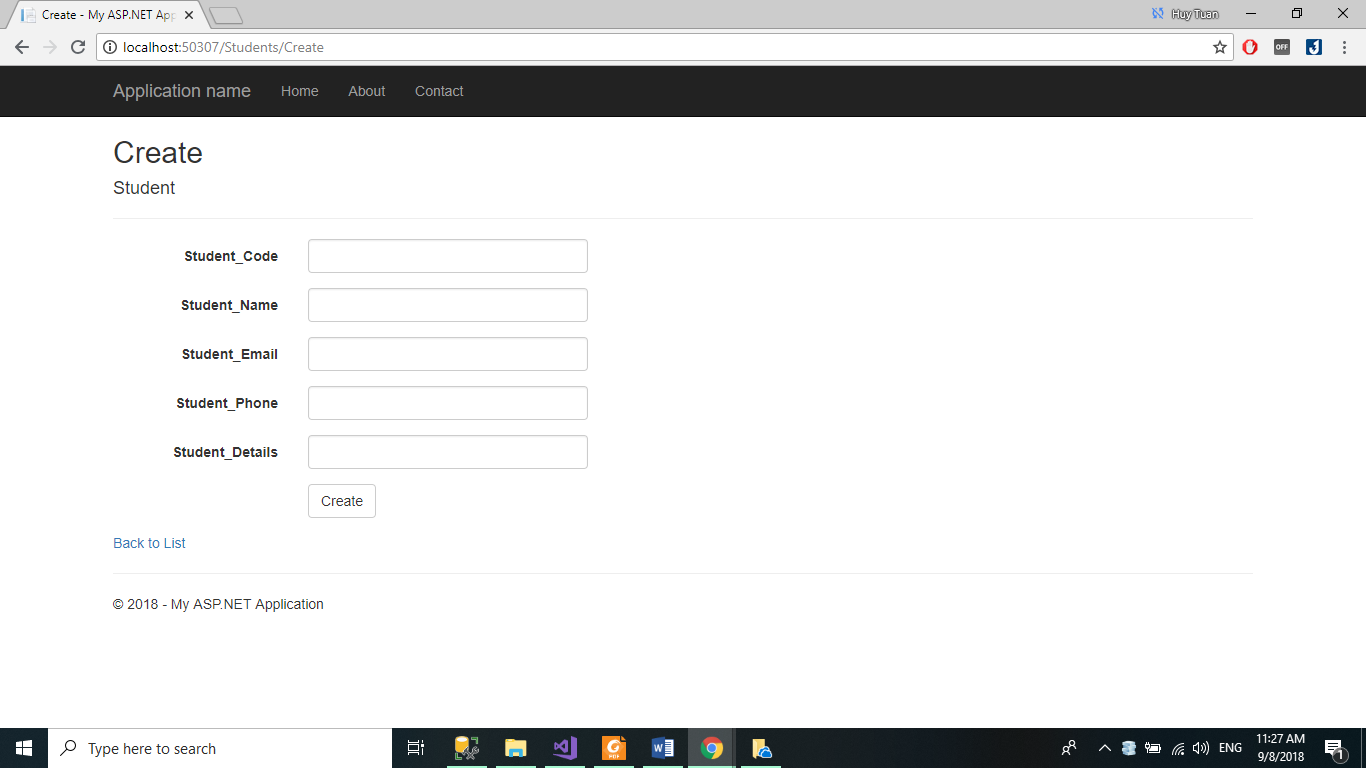
**Lưu ý: config để API chạy cùng với MVC**



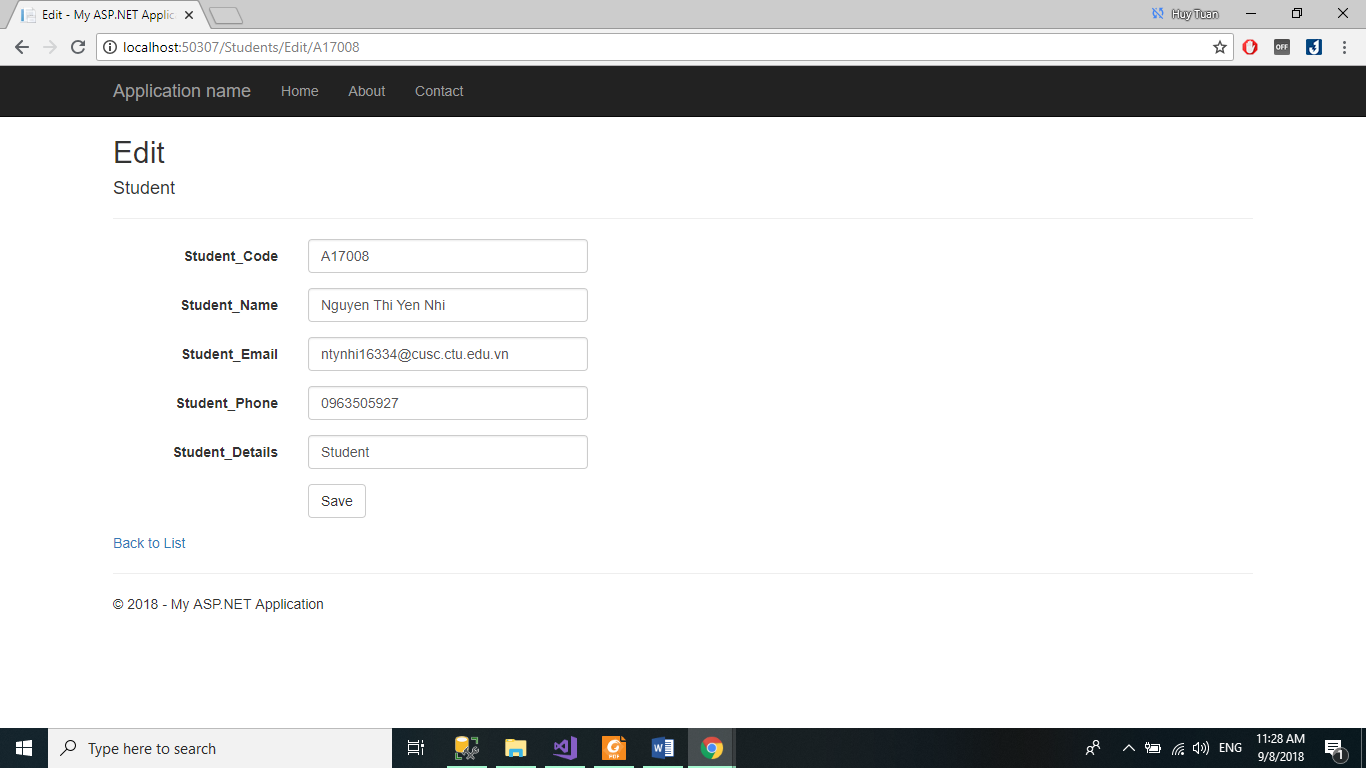
* **Index:**



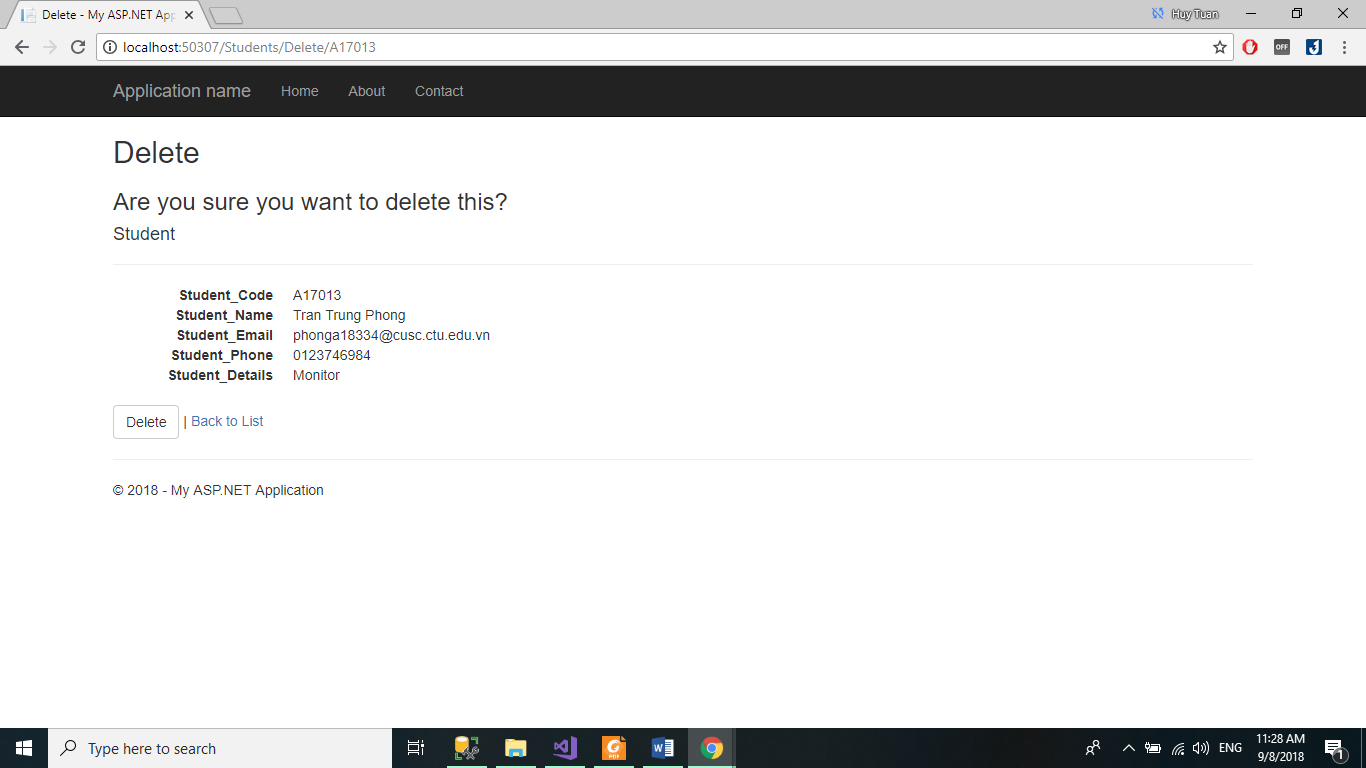
* **Create**



* **Edit**



* **Delete**



1. **Console – Client**
2. **WindowForm - Client**
3. **WCF**