**Integrate Entity Framework in MVC**

**Code First Approach**

**Step 1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication15.Models

{

public class Course

{

public int Id { get; set; }

public string Name { get; set; }

public string Batch { get; set; }

public int Duration { get; set; }

public string Subject { get; set; }

}

}

**Step 2: Make DBContext Class**

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Web;

namespace WebApplication15.Models

{

public class CourseDbContext : DbContext

{

public DbSet<Course> courses { get; set; }

}

}

**Step 3: Add Entry in config file for connectionString**

<connectionStrings>

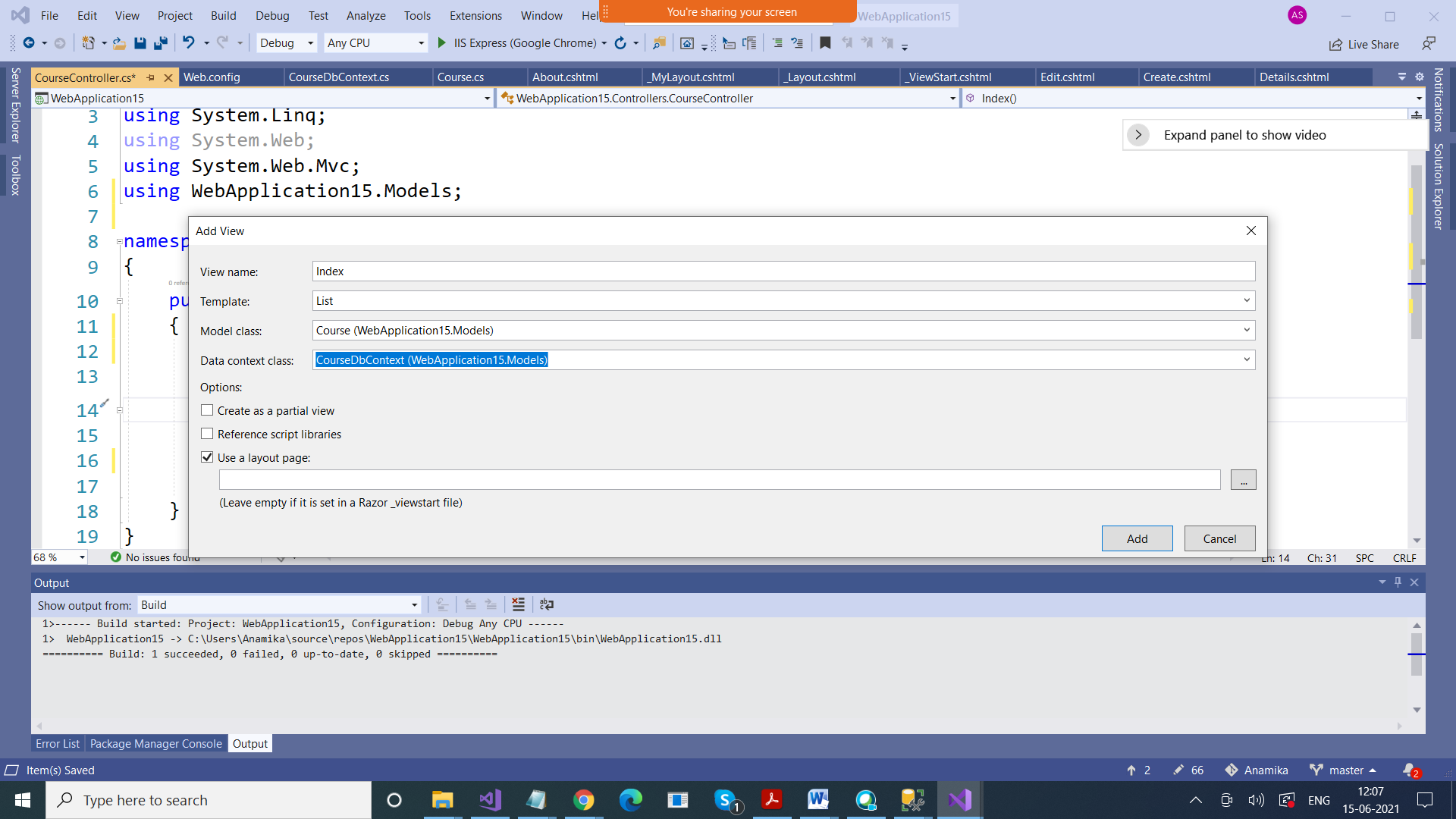
<add name="CourseDbContext" connectionString="data source= ;initial catalog= ; integrated security=true" providerName="System.Data.SqlClient"/>

</connectionStrings>

**Step 4 : Build your project**

**Step 5 : Add Controller**

**Perform CRUD Operations**



using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebApplication15.Models;

namespace WebApplication15.Controllers

{

public class CourseController : Controller

{

CourseDbContext db = new CourseDbContext();

// GET: Course

public ActionResult Index()

{

return View(db.courses.ToList());

}

public ActionResult Details(int? id)

{

if(id.HasValue)

{

Course course = db.courses.FirstOrDefault(x => x.Id == id);

if (course == null)

return View(course);

else

{

ViewBag.msg = "Course with this ID not found";

return View();

}

}

else

{

ViewBag.msg = "Please Provide ID";

return View();

}

}

public ActionResult Create()

{

Course course = new Course();

return View(course);

}

public ActionResult Create(Course course)

{

db.courses.Add(course);

db.SaveChanges();

TempData["msg"] ="Record inserted";

return RedirectToAction("Index");

}

}

}

---------------------------------------------------

Scaffolding Controllers alongwith Views

