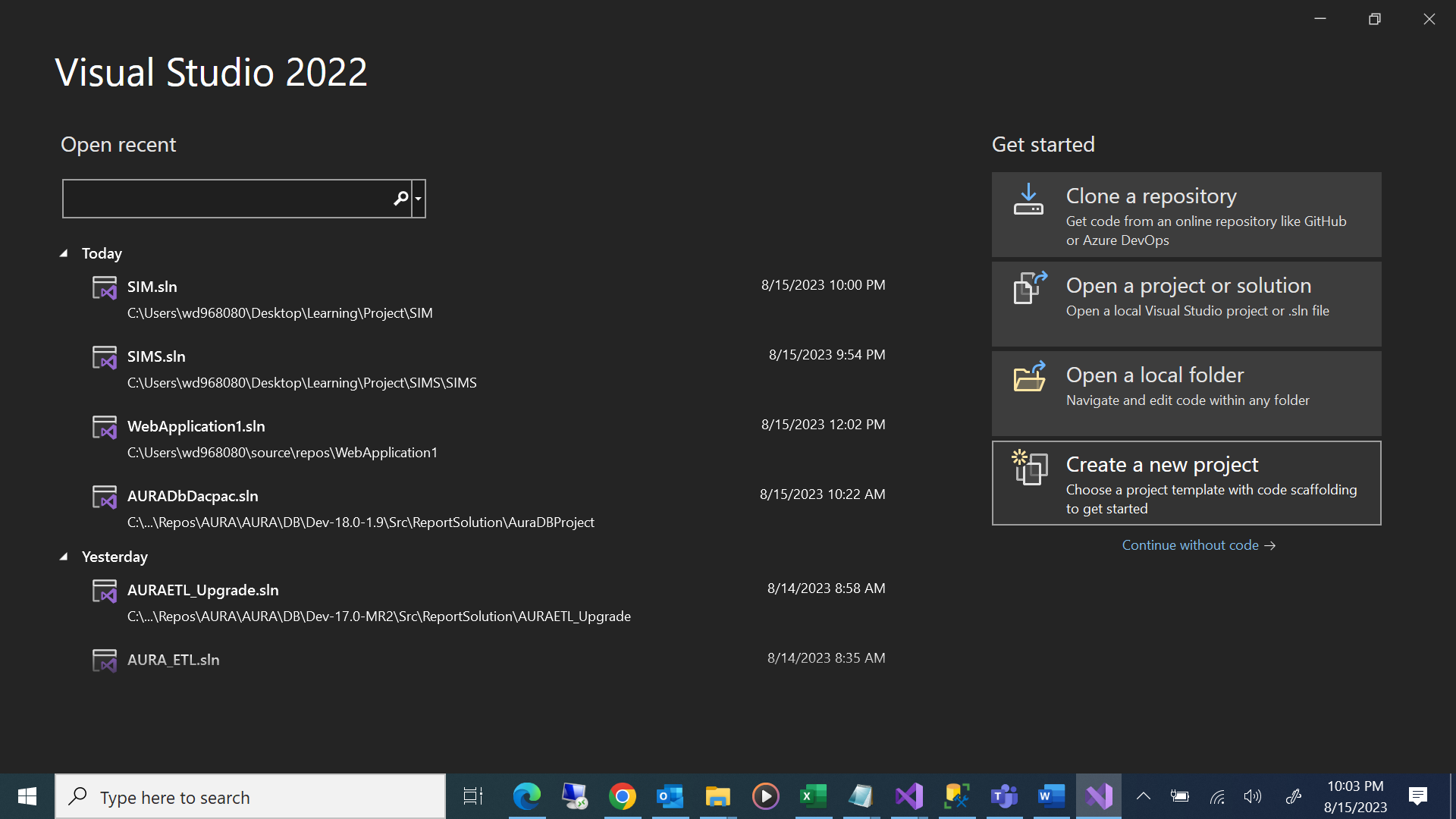
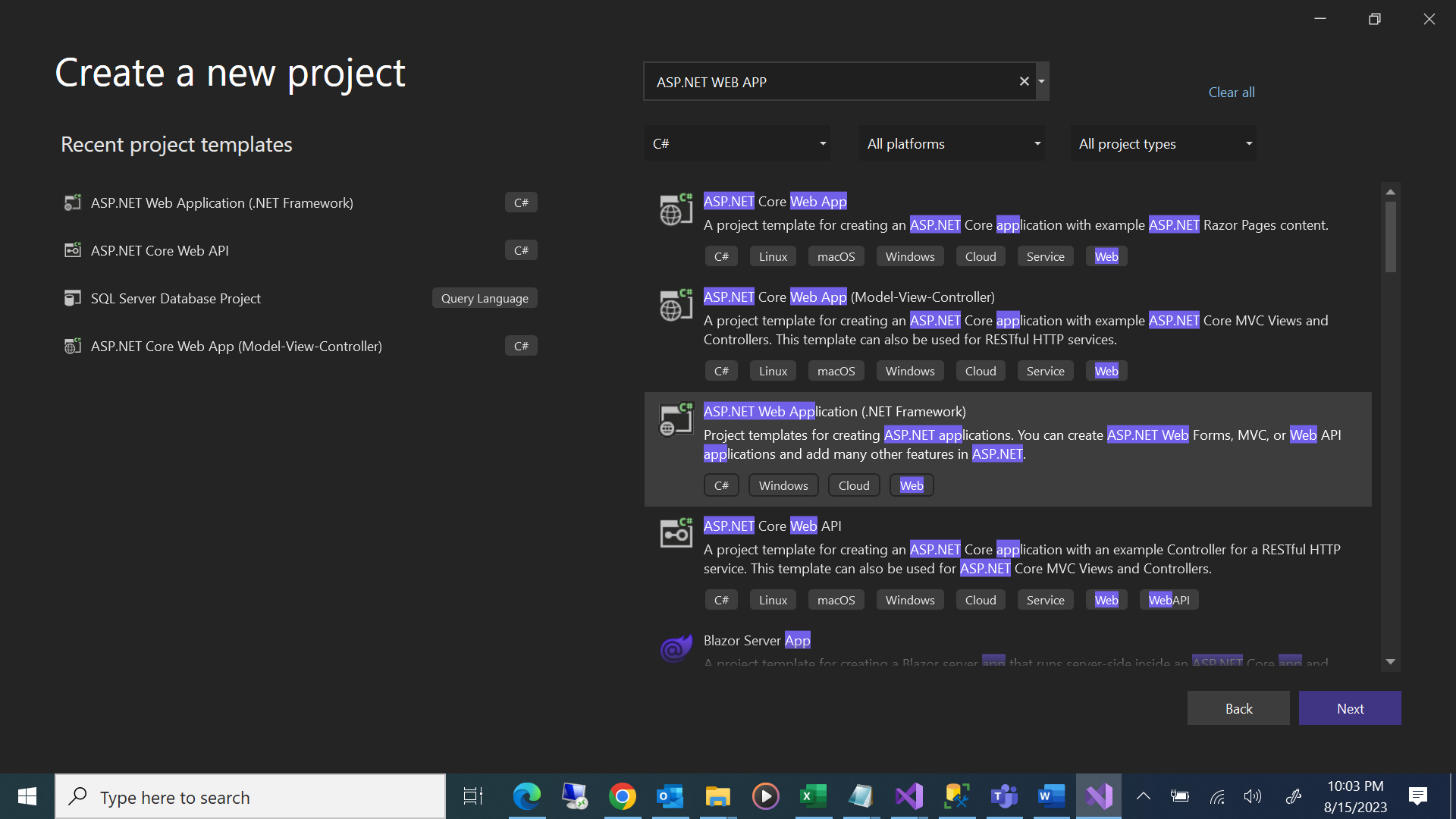
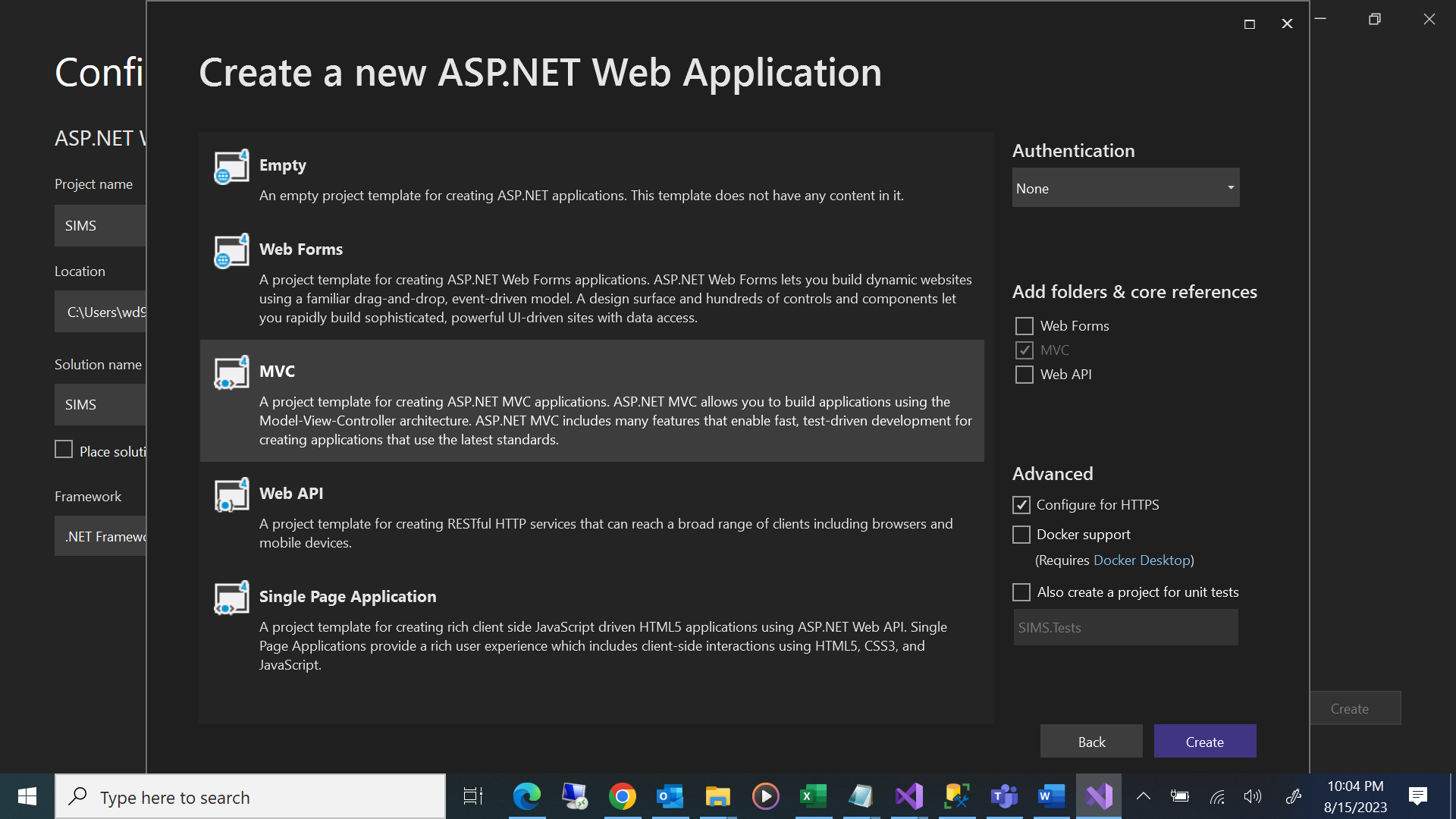
**How Do I?**

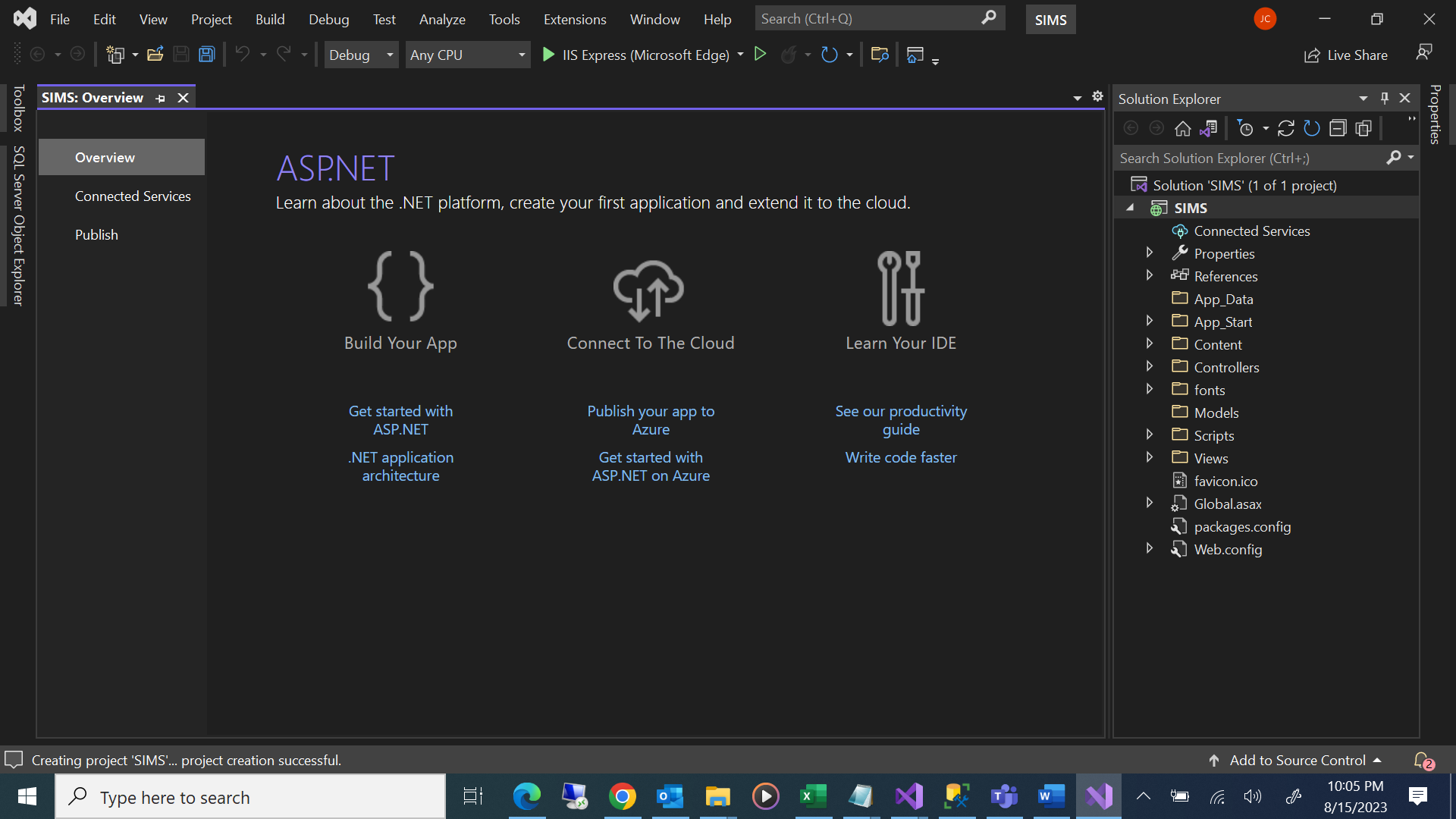


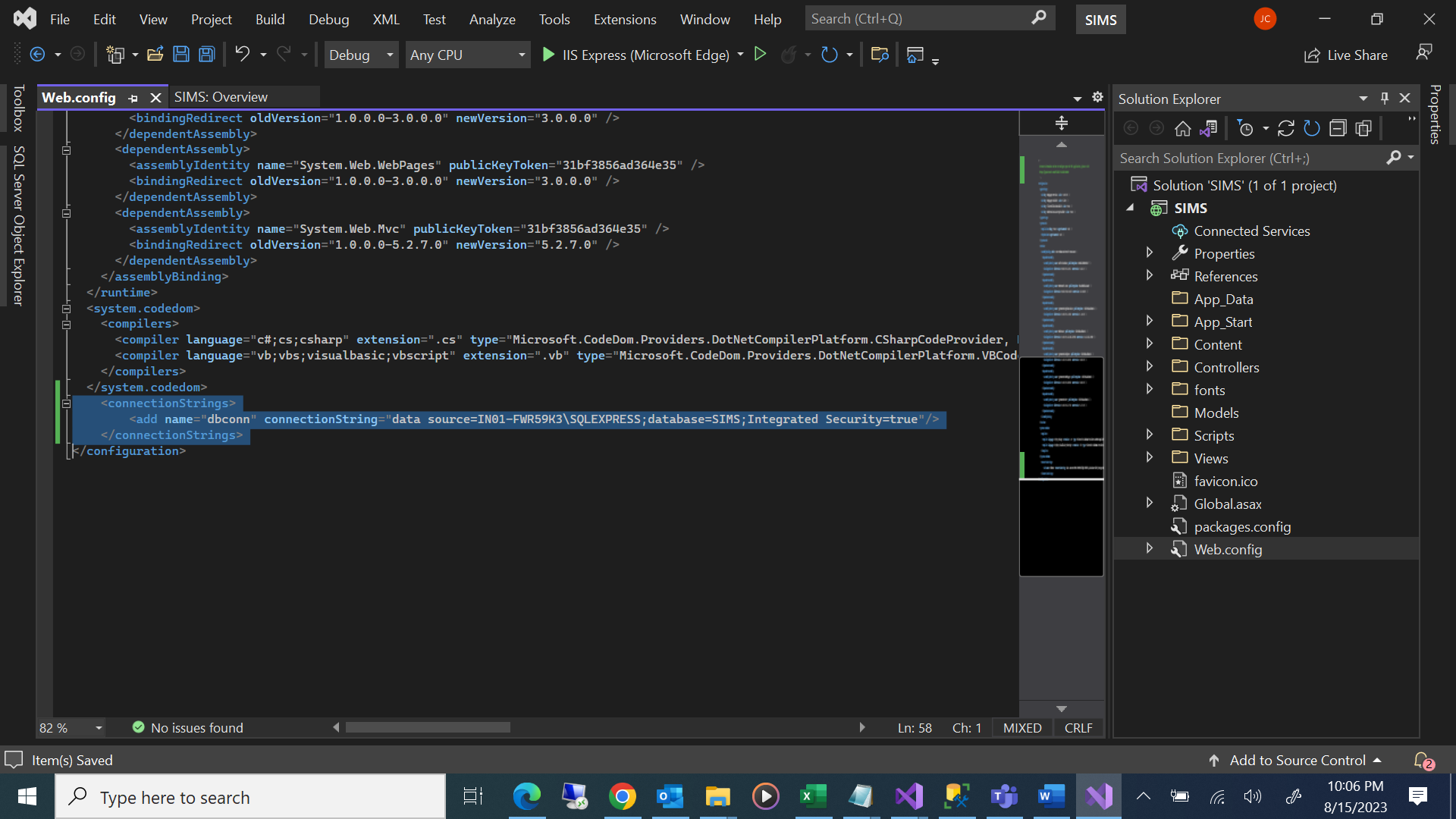


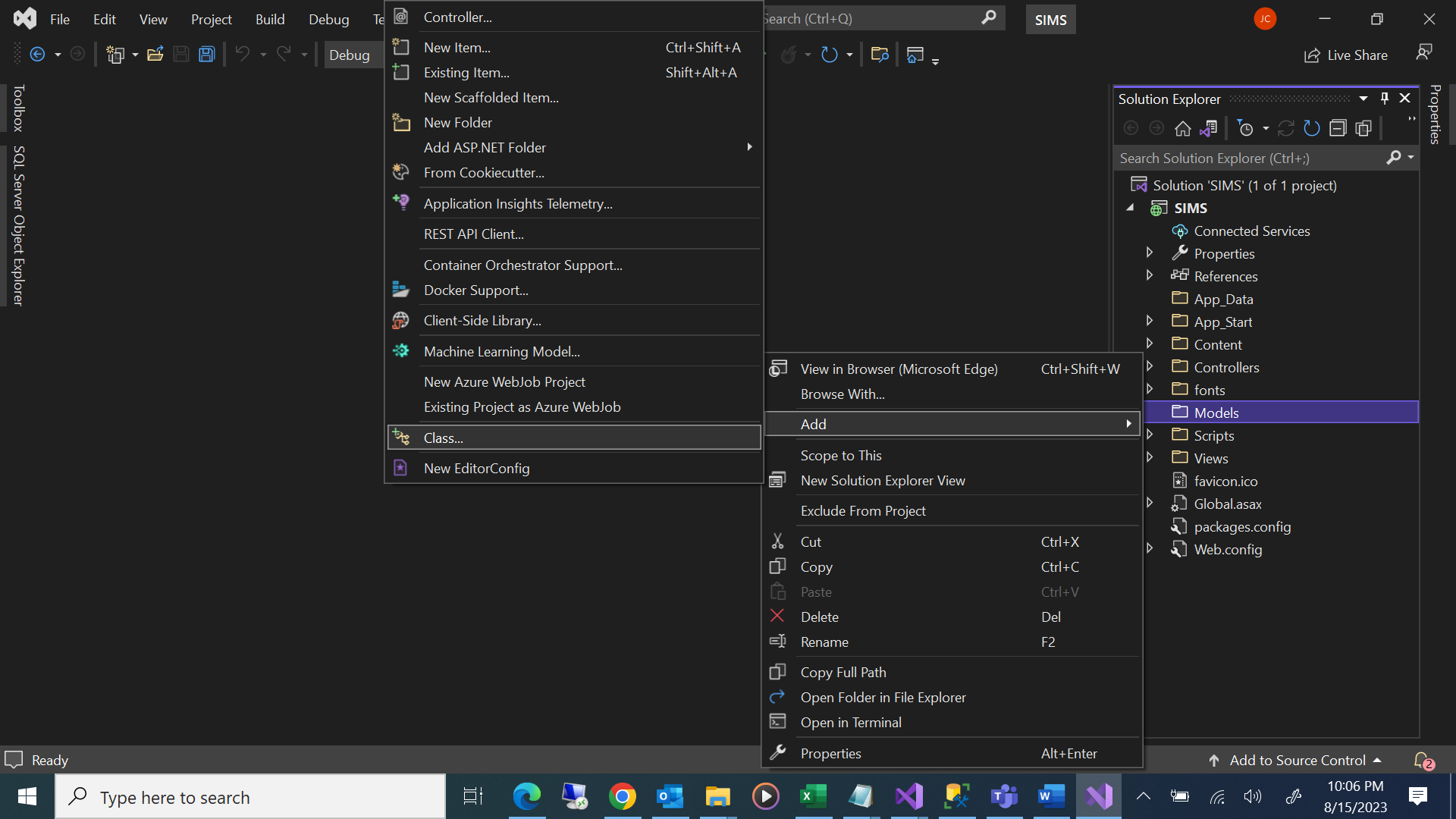


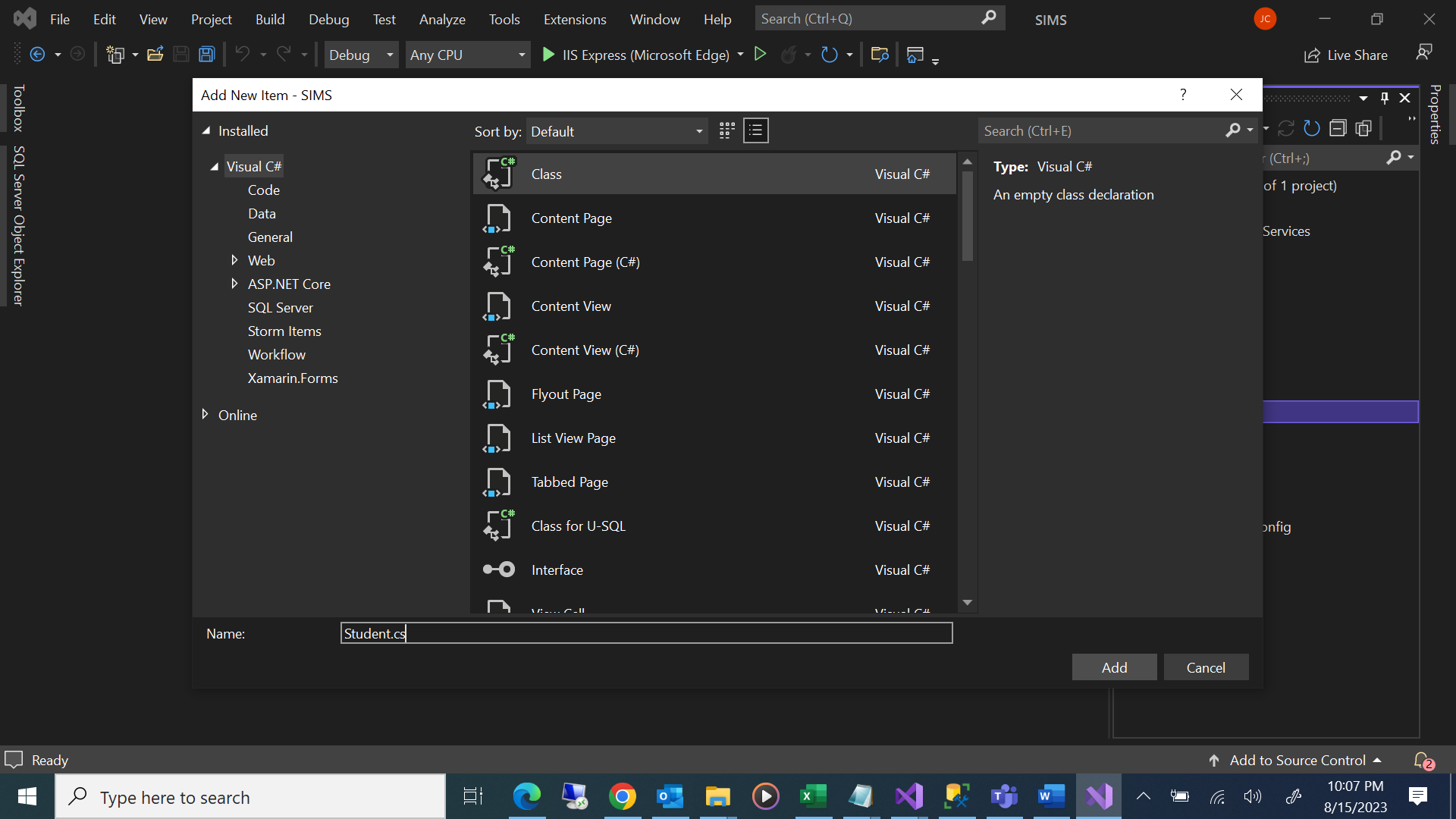


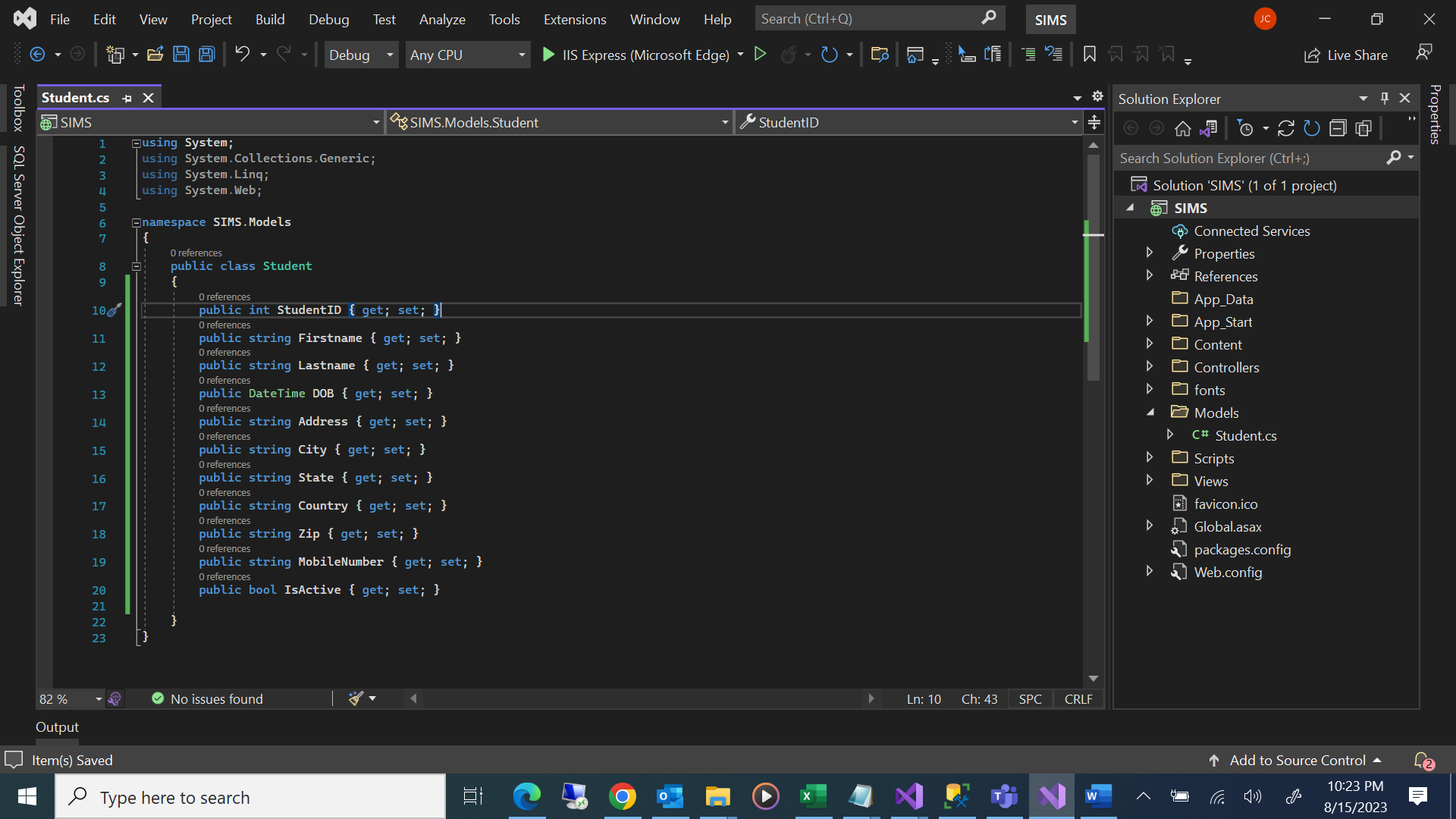












using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace SIMS.Models

{

public class Student

{

public int StudentID { get; set; }

public string Firstname { get; set; }

public string Lastname { get; set; }

public DateTime DOB { get; set; }

public string Address { get; set; }

public string City { get; set; }

public string State { get; set; }

public string Country { get; set; }

public string Zip { get; set; }

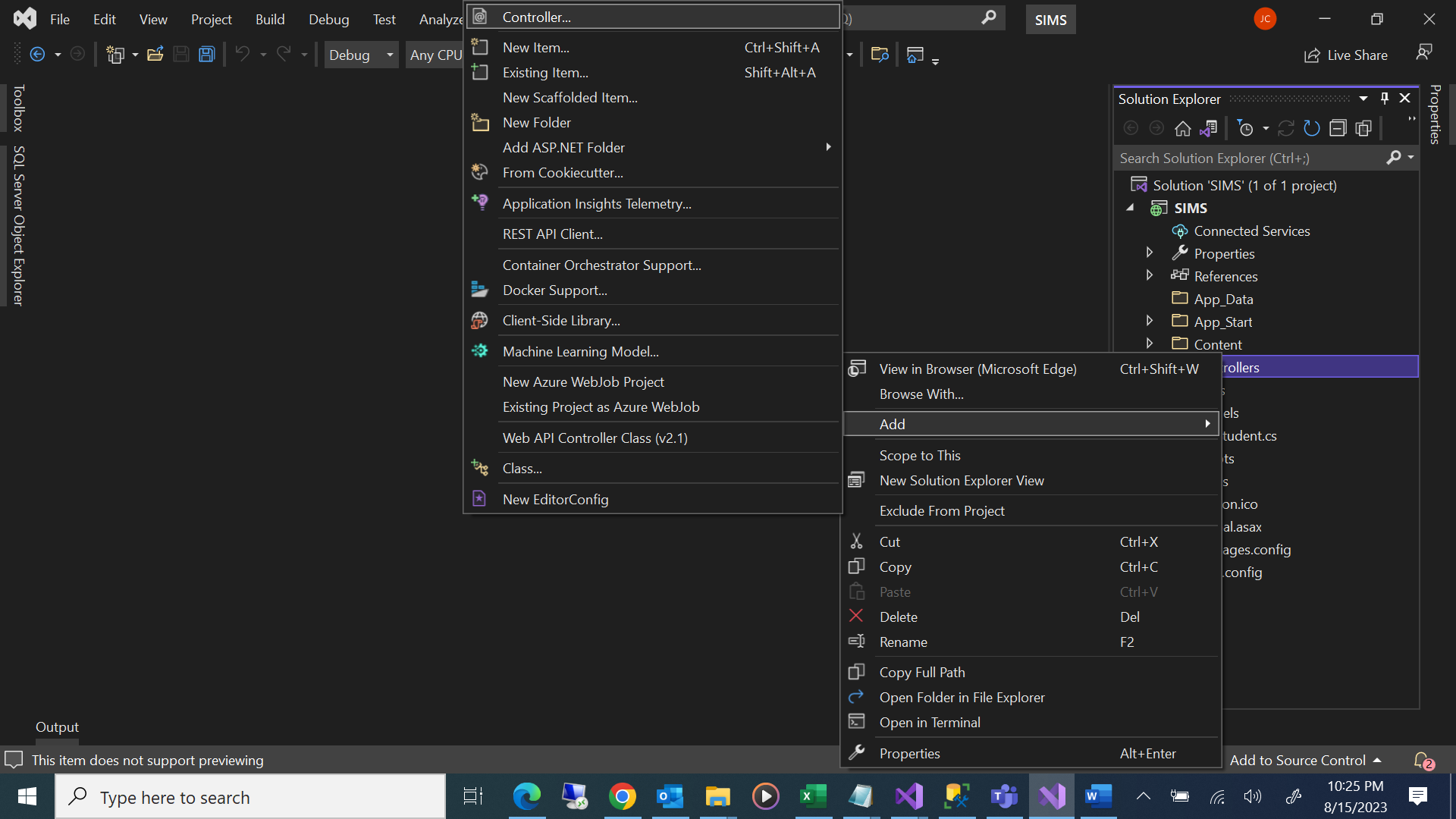
public string MobileNumber { get; set; }

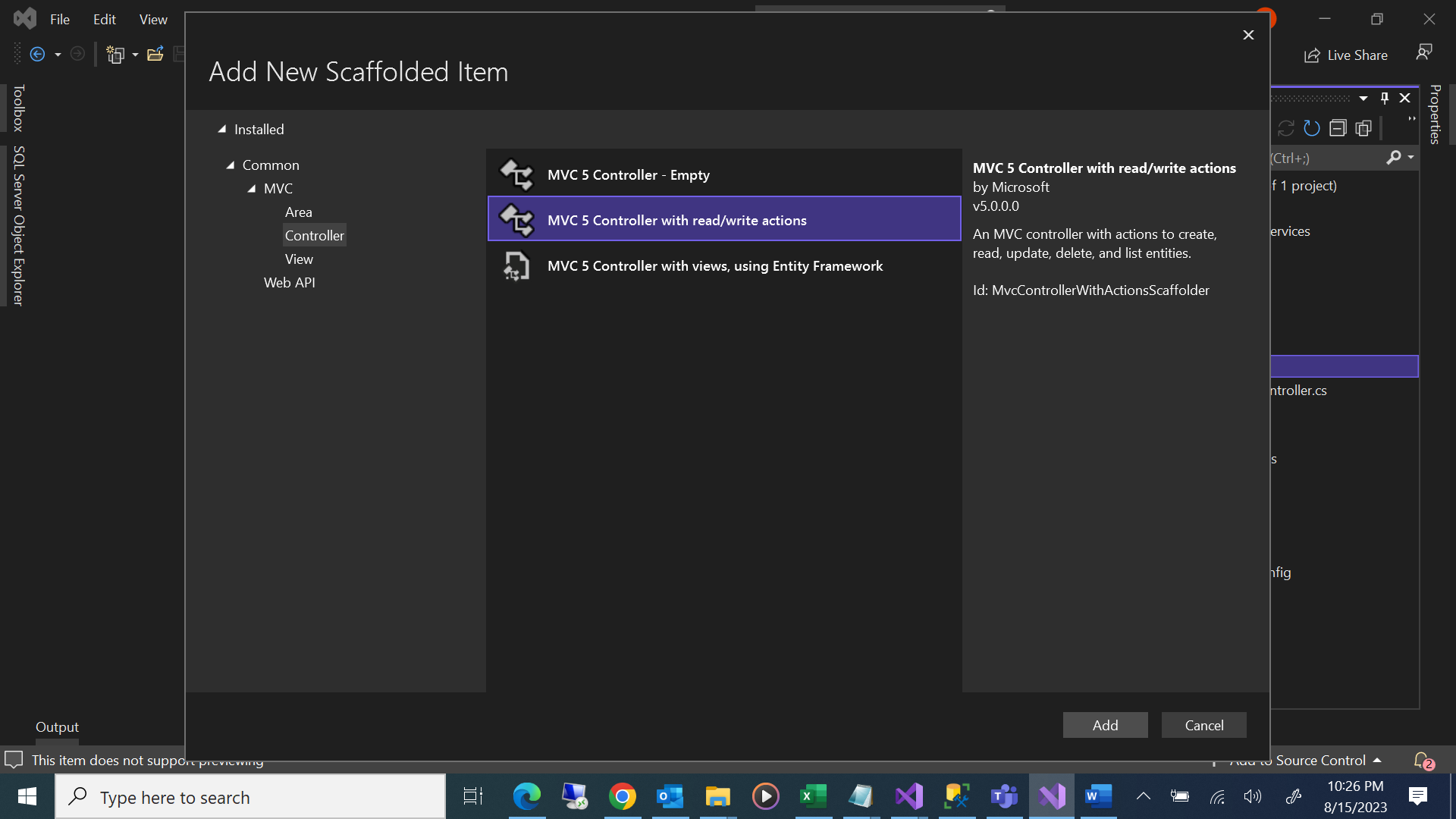
public bool IsActive { get; set; }

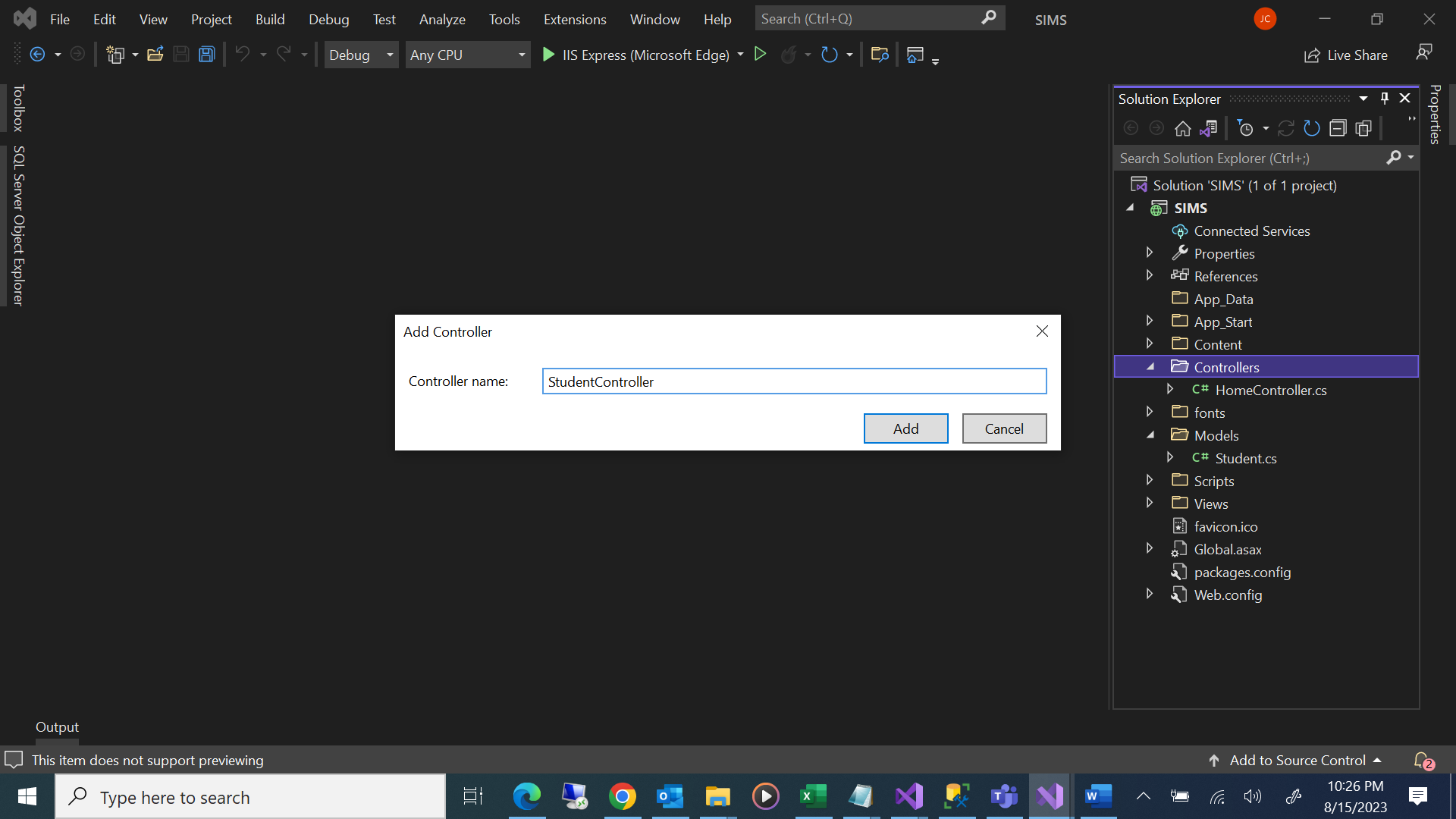
}

}

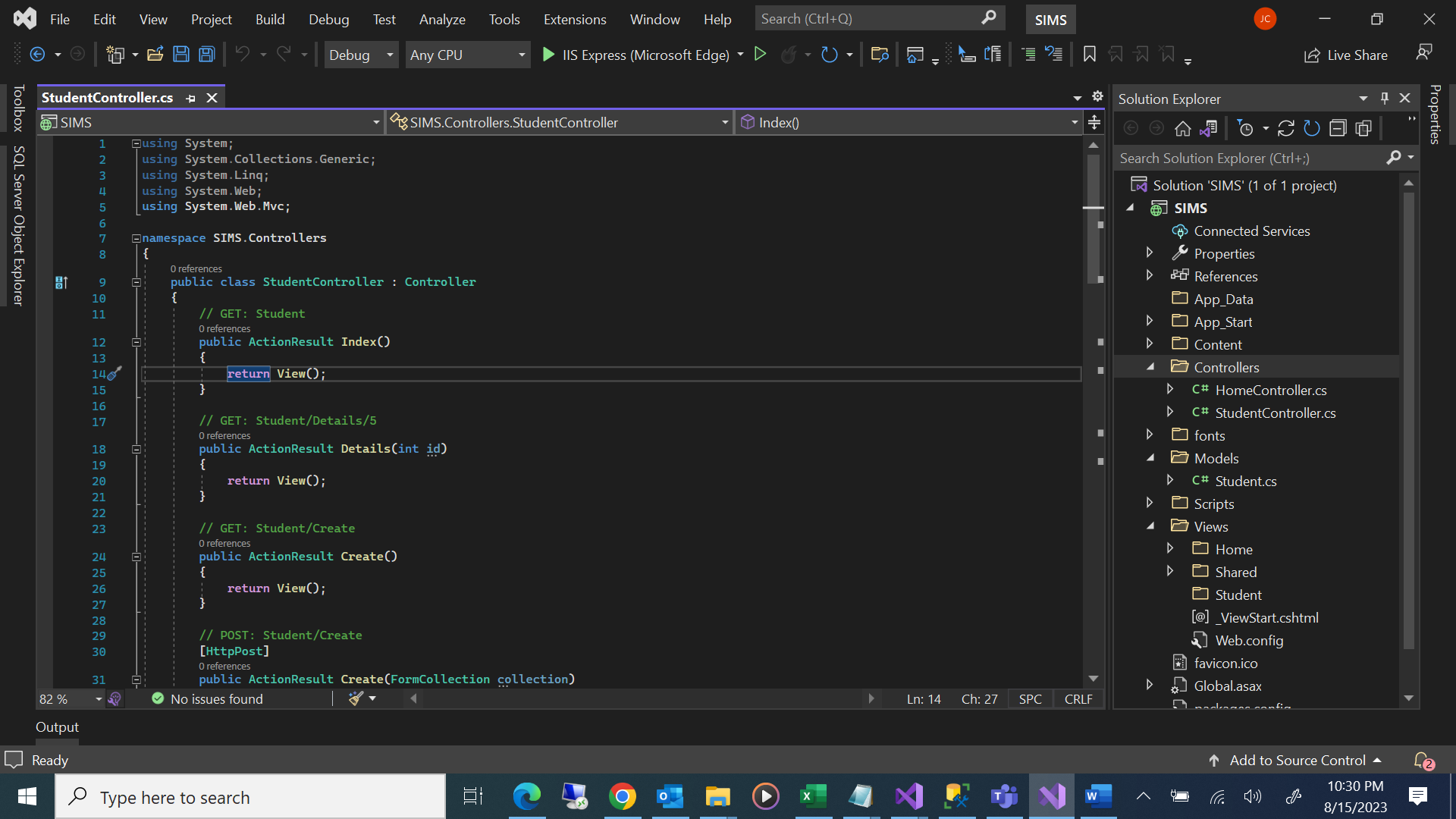
**Create Controller**

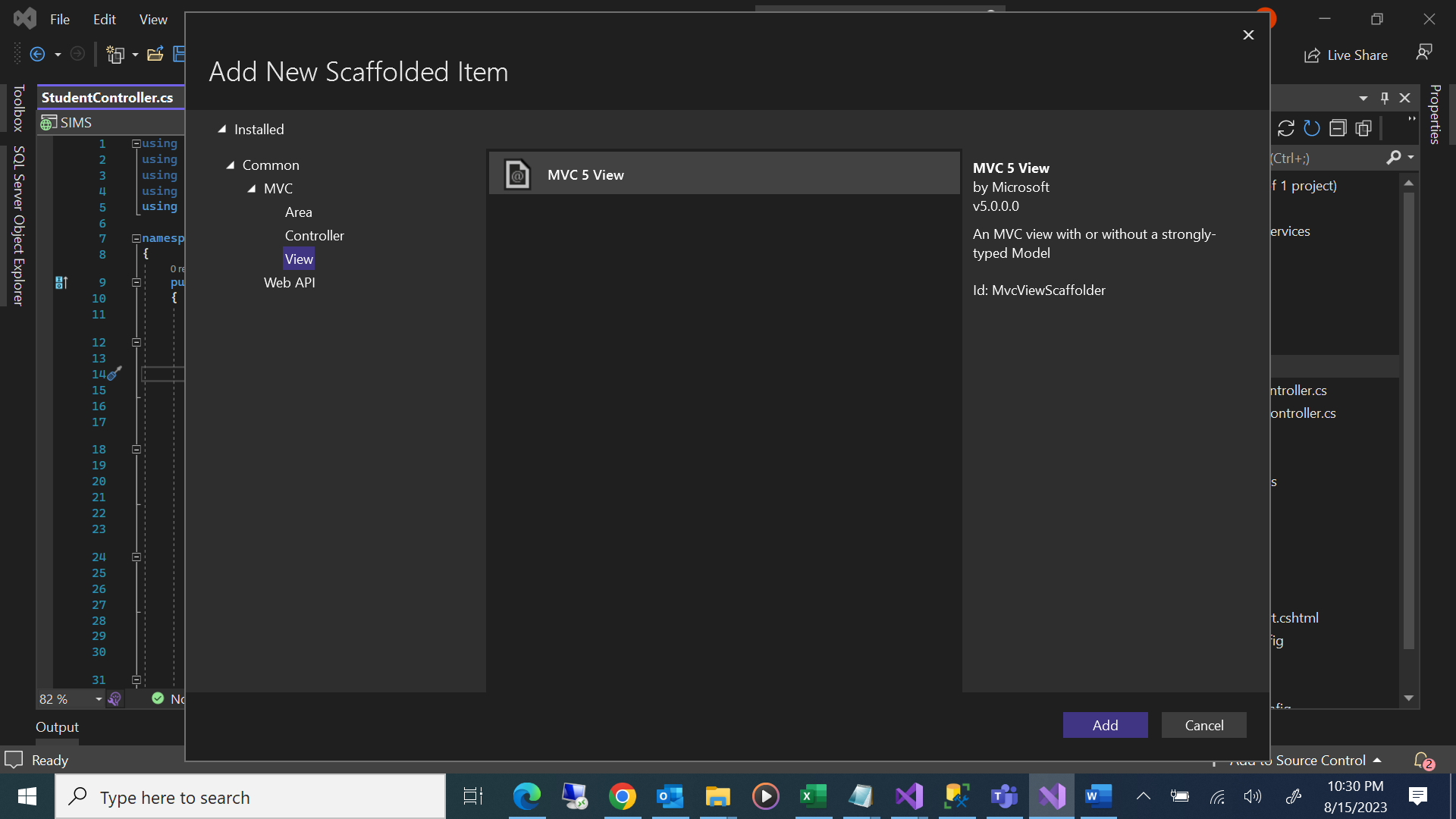


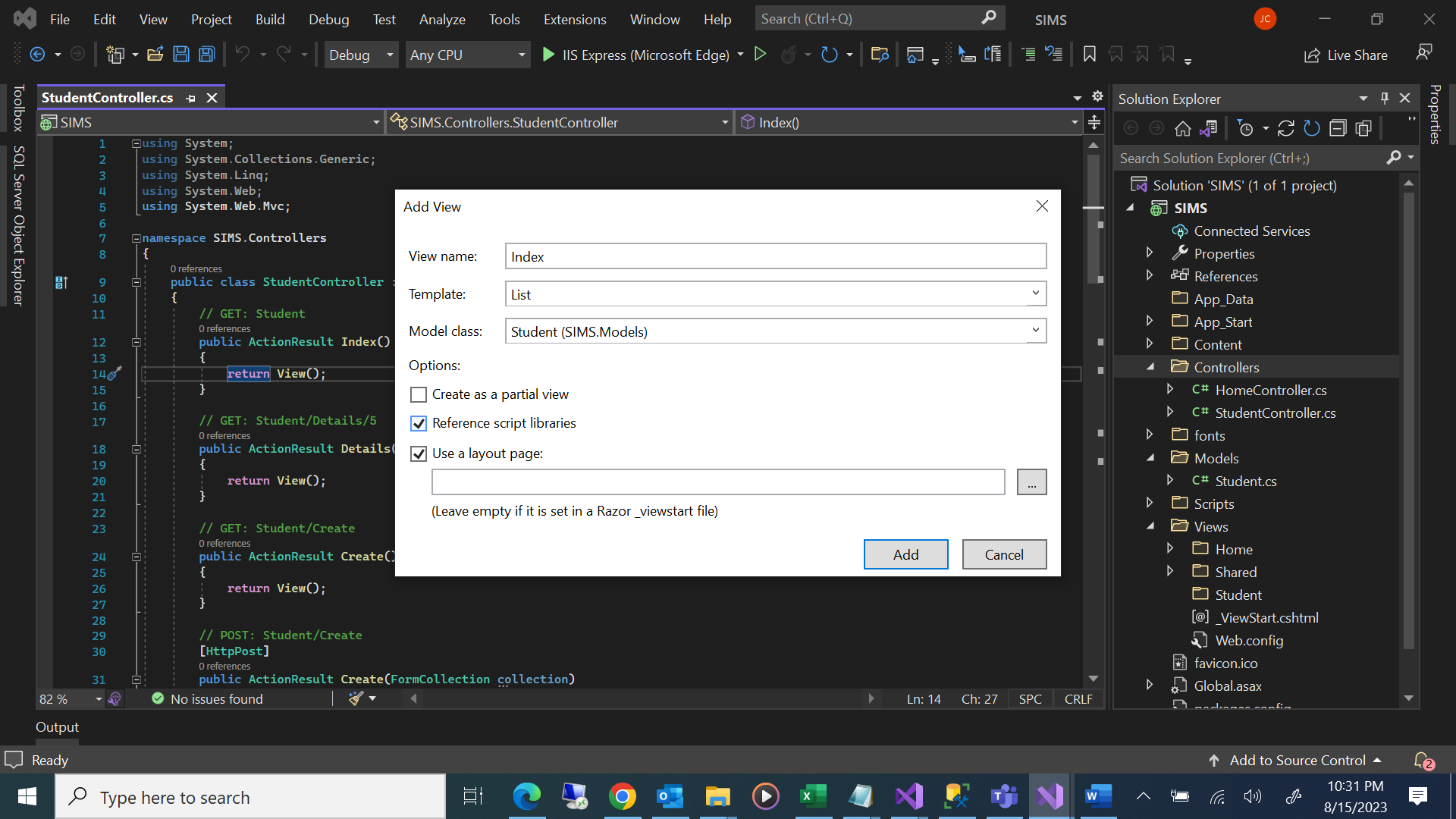


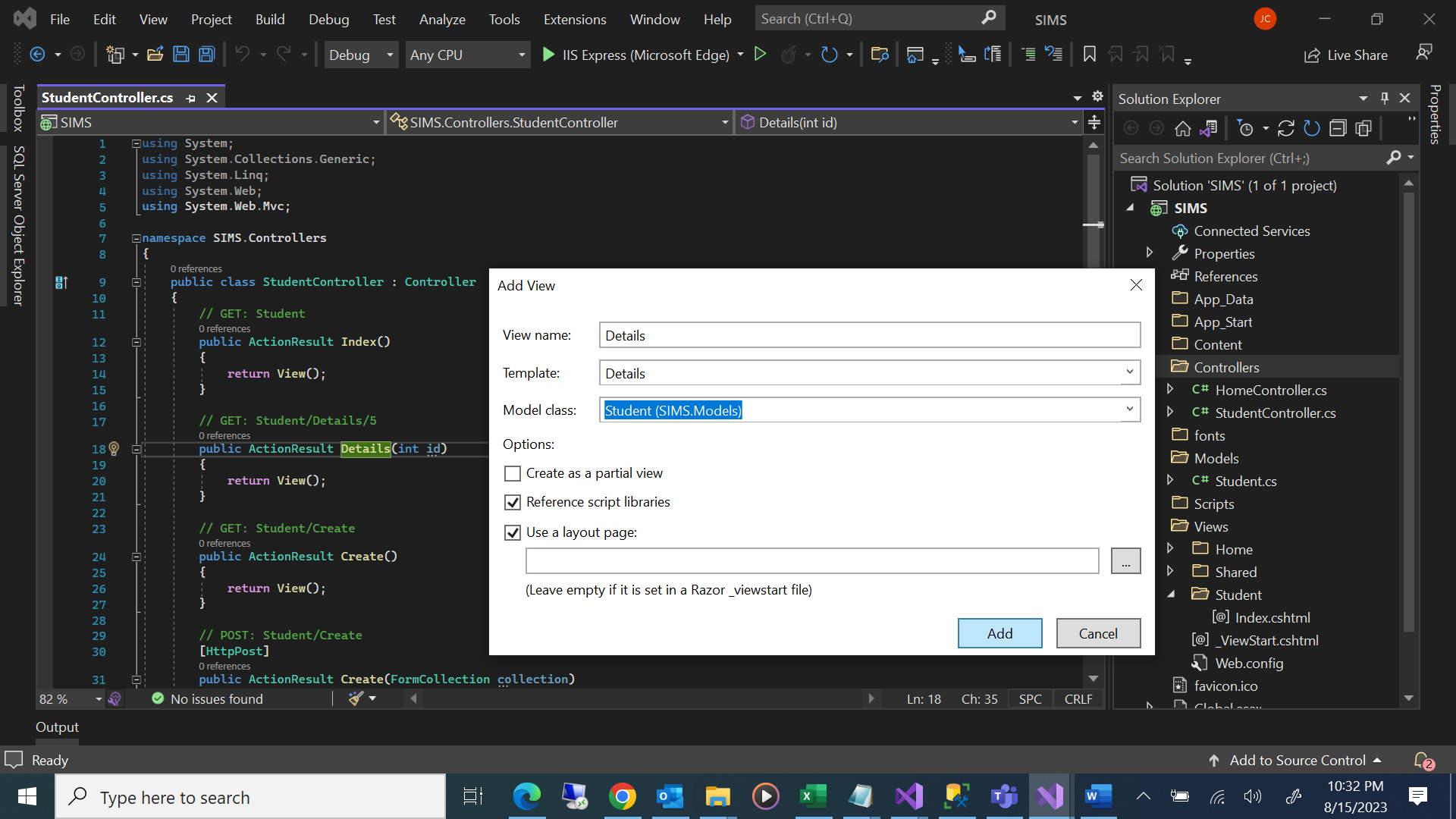


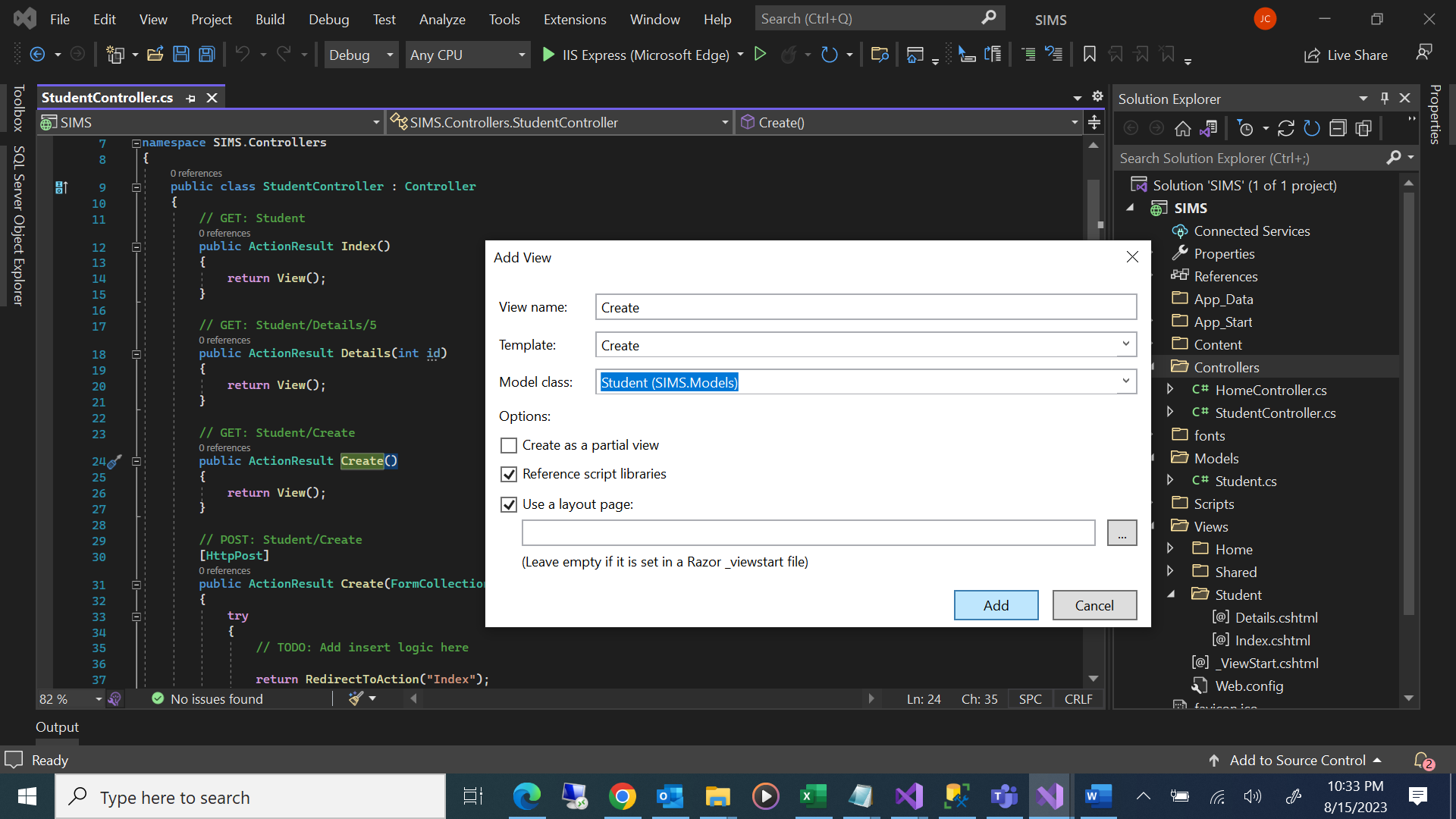
**Create View**

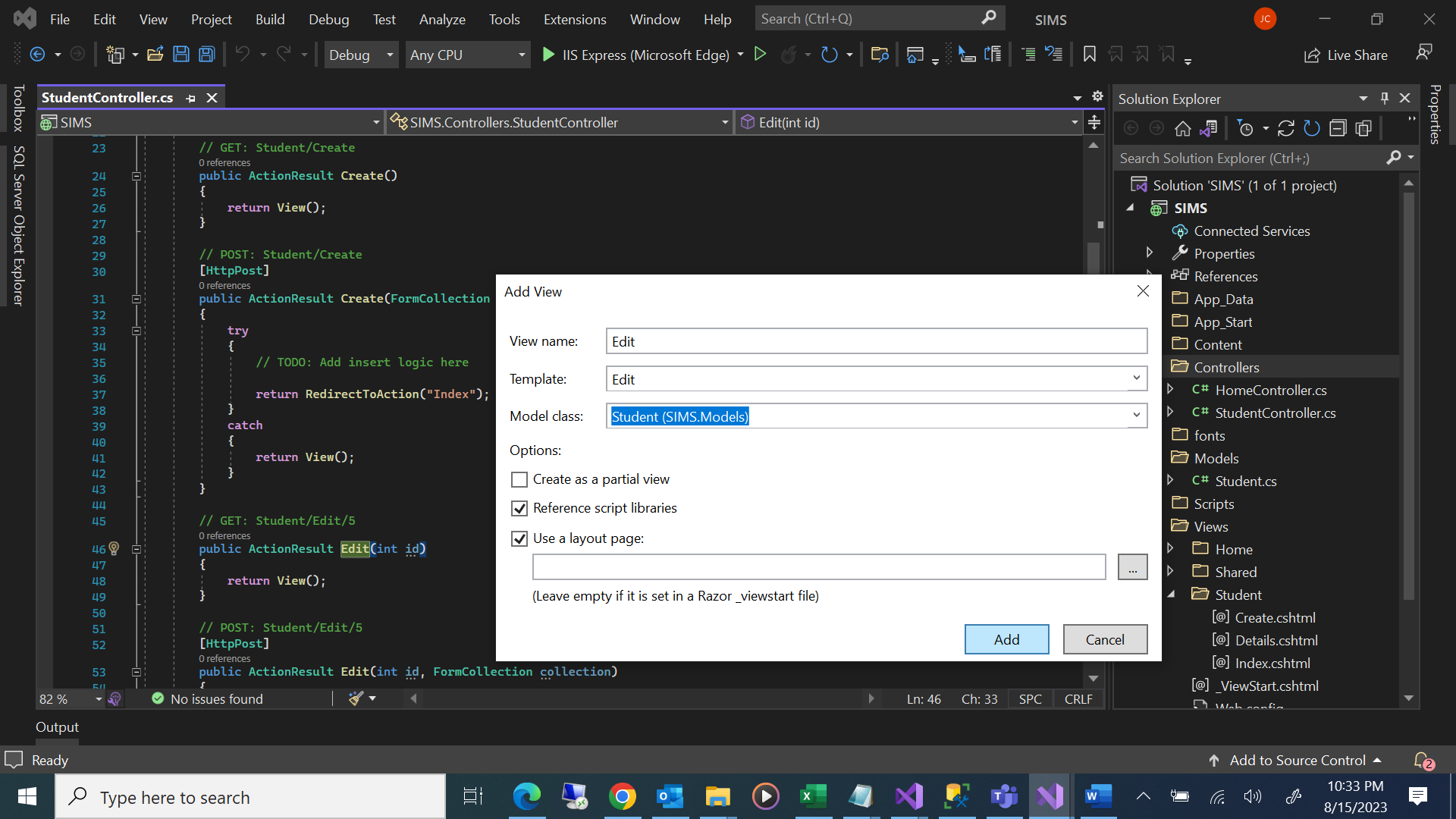


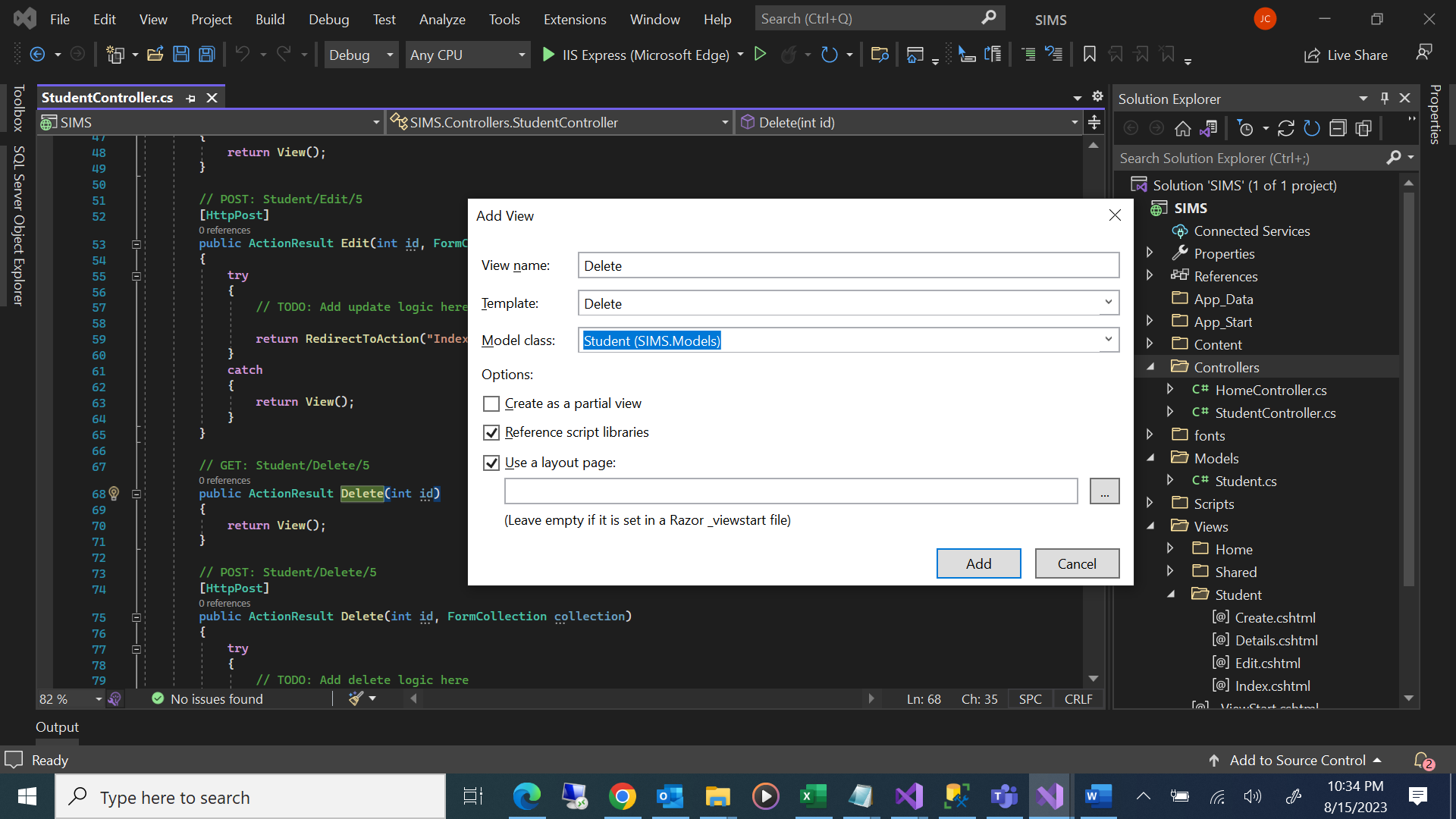




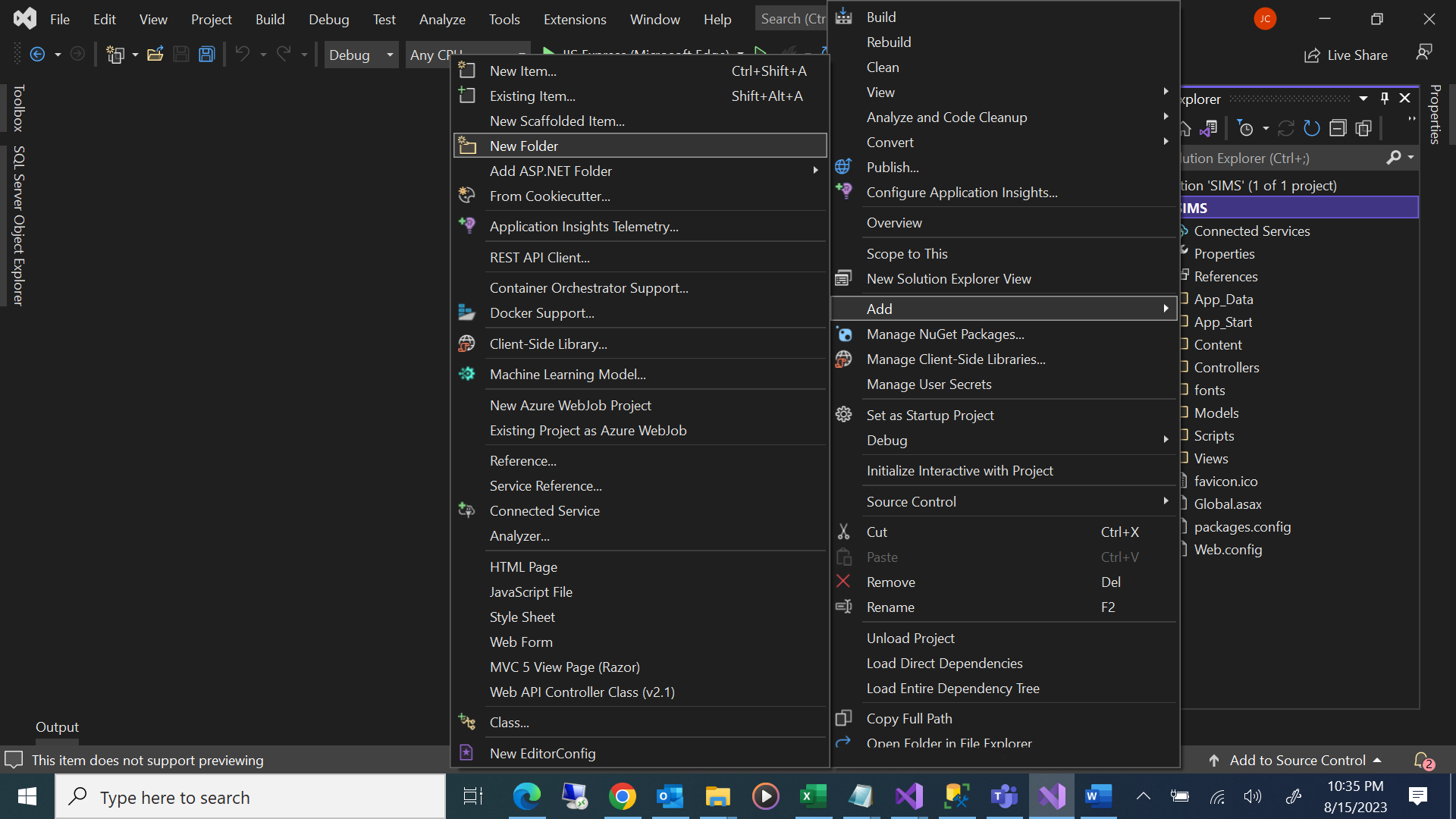




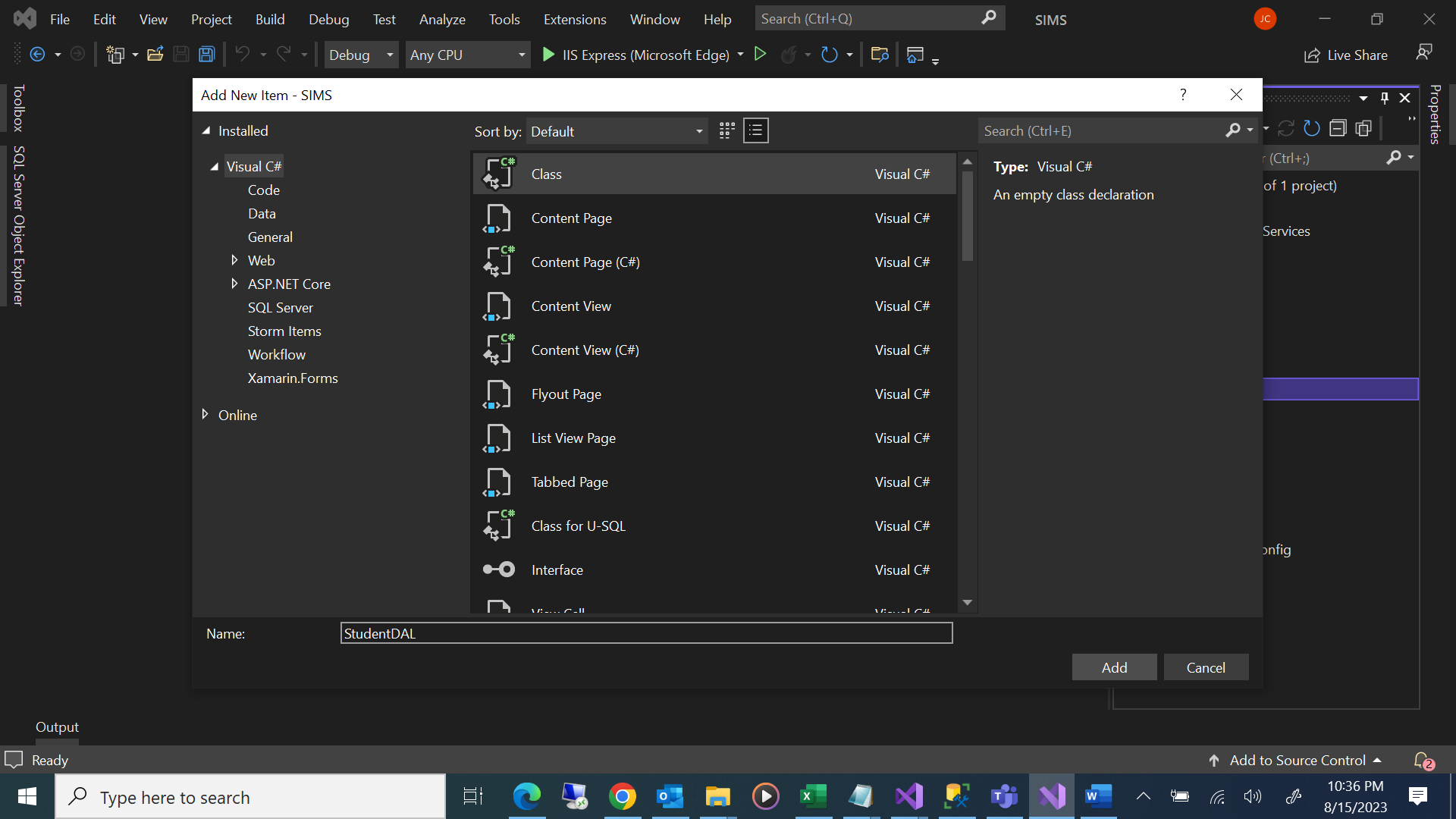




Create DAL







StudentDAL.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using SIMS.Models;

namespace SIMS.DAL

{

public class StudentDAL

{

string dbconnstring = ConfigurationManager.ConnectionStrings["dbconn"].ToString();

//Get all the students

public List<Student> GetAllStudents()

{

List<Student> studentList = new List<Student>();

using (SqlConnection connection = new SqlConnection(dbconnstring))

{

SqlCommand command = connection.CreateCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "sproc\_GetAllStudent";

SqlDataAdapter sqlDA = new SqlDataAdapter(command);

DataTable dtStudent = new DataTable();

connection.Open();

sqlDA.Fill(dtStudent);

connection.Close();

foreach (DataRow dr in dtStudent.Rows)

{

studentList.Add(new Student

{

StudentID = Convert.ToInt32(dr["StudentID"]),

Firstname = dr["Firstname"].ToString(),

Lastname = dr["LastName"].ToString(),

DOB = Convert.ToDateTime(dr["DOB"]),

Address = dr["Address"].ToString(),

City = dr["City"].ToString(),

State = dr["State"].ToString(),

Country = dr["Country"].ToString(),

Zip = dr["Zip"].ToString(),

MobileNumber = dr["MobileNumber"].ToString(),

IsActive = Convert.ToBoolean(dr["IsActive"])

});

}

}

return studentList;

}

//Insert students

public bool InsertStudents(Student student)

{

using (SqlConnection connection = new SqlConnection(dbconnstring))

{

SqlCommand command = new SqlCommand("sproc\_InsertStudent", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@Firstname", student.Firstname);

command.Parameters.AddWithValue("@LastName", student.Lastname);

command.Parameters.AddWithValue("@DOB", student.DOB);

command.Parameters.AddWithValue("@Address", student.Address);

command.Parameters.AddWithValue("@City", student.City);

command.Parameters.AddWithValue("@State", student.State);

command.Parameters.AddWithValue("@Country", student.Country);

command.Parameters.AddWithValue("@Zip", student.Zip);

command.Parameters.AddWithValue("@MobileNumber", student.MobileNumber);

command.Parameters.AddWithValue("@IsActive", student.IsActive);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

return true;

}

//Get Student By ID

public List<Student> GetStudentByID(int StudentID)

{

List<Student> studentList = new List<Student>();

using (SqlConnection connection = new SqlConnection(dbconnstring))

{

SqlCommand command = connection.CreateCommand();

command.CommandType = CommandType.StoredProcedure;

command.CommandText = "sproc\_GetStudentByID";

command.Parameters.AddWithValue("@StudentID", StudentID);

SqlDataAdapter sqlDA = new SqlDataAdapter(command);

DataTable dtStudent = new DataTable();

connection.Open();

sqlDA.Fill(dtStudent);

connection.Close();

foreach (DataRow dr in dtStudent.Rows)

{

studentList.Add(new Student

{

StudentID = Convert.ToInt32(dr["StudentID"]),

Firstname = dr["Firstname"].ToString(),

Lastname = dr["LastName"].ToString(),

DOB = Convert.ToDateTime(dr["DOB"]),

Address = dr["Address"].ToString(),

City = dr["City"].ToString(),

State = dr["State"].ToString(),

Country = dr["Country"].ToString(),

Zip = dr["Zip"].ToString(),

MobileNumber = dr["MobileNumber"].ToString(),

IsActive = Convert.ToBoolean(dr["IsActive"])

});

}

}

return studentList;

}

//Update Student By ID

public bool UpdateStudents(Student student)

{

using (SqlConnection connection = new SqlConnection(dbconnstring))

{

SqlCommand command = new SqlCommand("sproc\_UpdateStudentByID", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@StudentID", student.StudentID);

command.Parameters.AddWithValue("@Firstname", student.Firstname);

command.Parameters.AddWithValue("@LastName", student.Lastname);

command.Parameters.AddWithValue("@DOB", student.DOB);

command.Parameters.AddWithValue("@Address", student.Address);

command.Parameters.AddWithValue("@City", student.City);

command.Parameters.AddWithValue("@State", student.State);

command.Parameters.AddWithValue("@Country", student.Country);

command.Parameters.AddWithValue("@Zip", student.Zip);

command.Parameters.AddWithValue("@MobileNumber", student.MobileNumber);

command.Parameters.AddWithValue("@IsActive", student.IsActive);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

return true;

}

//Delete Student By ID

public bool DeleteStudents(int StuentID)

{

using (SqlConnection connection = new SqlConnection(dbconnstring))

{

SqlCommand command = new SqlCommand("sproc\_DeleteStudentByID", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@StudentID", StuentID);

connection.Open();

command.ExecuteNonQuery();

connection.Close();

}

return true;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using SIMS.DAL;

using SIMS.Models;

namespace SIMS.Controllers

{

public class StudentController : Controller

{

StudentDAL ObjStudentDAL = new StudentDAL();

// GET: Student

public ActionResult Index()

{

var studentlist = ObjStudentDAL.GetAllStudents();

return View(studentlist);

}

// GET: Student/Details/5

public ActionResult Details(int id)

{

var student = ObjStudentDAL.GetStudentByID(id).FirstOrDefault();

return View(student);

}

// GET: Student/Create

public ActionResult Create()

{

return View();

}

// POST: Student/Create

[HttpPost]

public ActionResult Create(Student ObjStudent)

{

try

{

// TODO: Add insert logic here

ObjStudentDAL.InsertStudents(ObjStudent);

return RedirectToAction("Index");

}

catch

{

return View();

}

}

// GET: Student/Edit/5

public ActionResult Edit(int id)

{

var student = ObjStudentDAL.GetStudentByID(id).FirstOrDefault();

return View(student);

}

// POST: Student/Edit/5

[HttpPost, ActionName("Edit")]

public ActionResult Edit(Student ObjStudent)

{

try

{

// TODO: Add update logic here

ObjStudentDAL.UpdateStudents(ObjStudent);

return RedirectToAction("Index");

}

catch

{

return View();

}

}

// GET: Student/Delete/5

public ActionResult Delete(int id)

{

var student = ObjStudentDAL.GetStudentByID(id).FirstOrDefault();

return View(student);

}

// POST: Student/Delete/5

[HttpPost, ActionName("Delete")]

public ActionResult DeleteStudent(int id)

{

try

{

// TODO: Add delete logic here

ObjStudentDAL.DeleteStudents(id);

return RedirectToAction("Index");

}

catch

{

return View();

}

}

}

}