Student Management System

Here we have used the MVC with the N-tier architecture which includes MVC web application (Presentation layer), Database Layer and Business Layer.

MVC Layer (Presentation Layer) – to display the frontend using views

Business layer – To set the logic and connect between the database layer and presentation layer

Database layer – To handle all database queries (We have created a database using Entity Framework)

                           Following are the Tables we have used

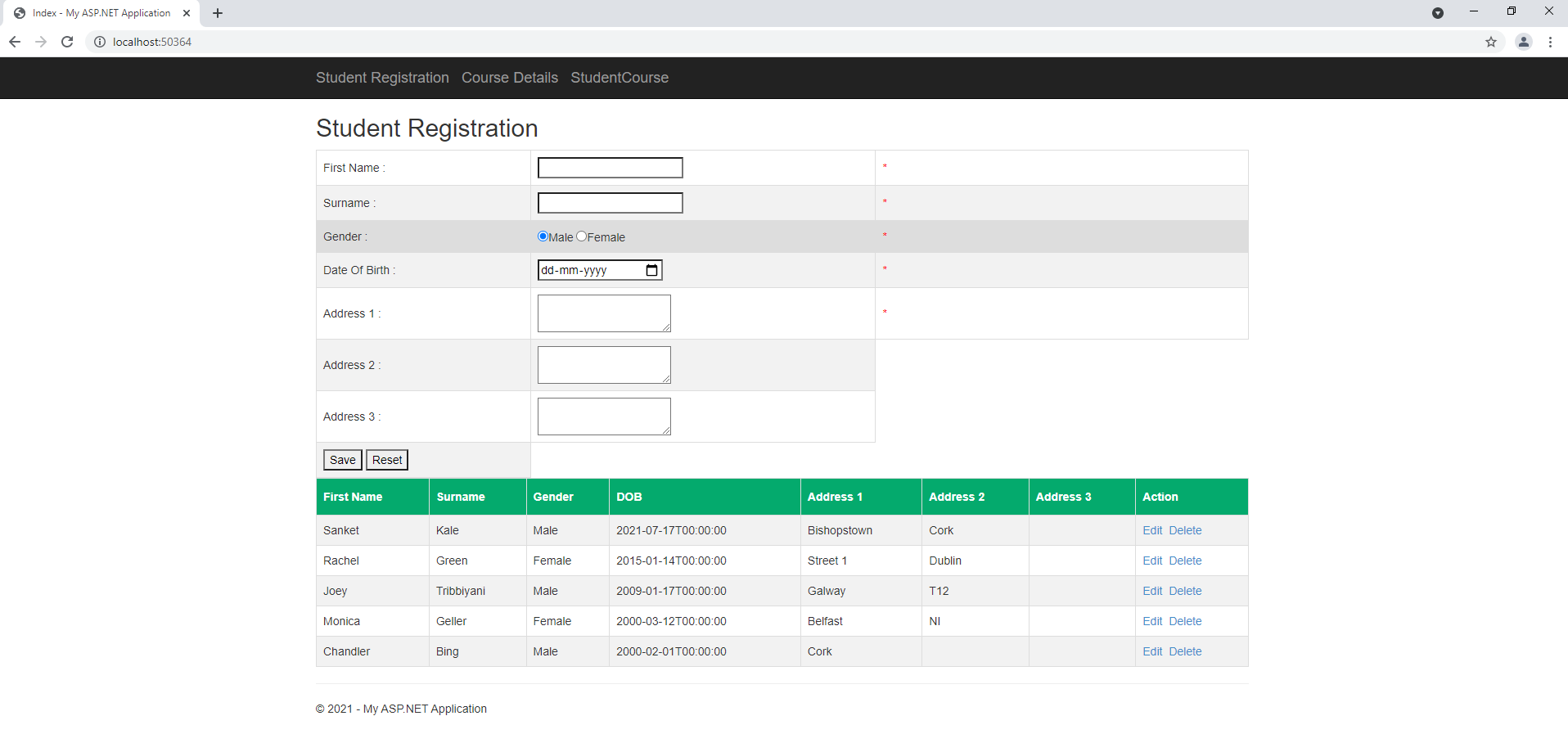
* tblStudent - student Details
* tblCourse - course Details
* tblStudentCourse – Add mapping for student (studentid) and course (CourseCode)
* tblCourseCount - To manage total seats for each course

 We split the application into N-tier architecture, which isolates each layer and allows developers to work on each layer independently. This design is beneficial for long-term software maintenance.

 Following are the specification of the application

Student Registration

* Controller – HomeController
* View – Home/courseIndex.cshtml
* Business Layer – StudentModel.cs
* Database Layer – StudentRegistration.cs

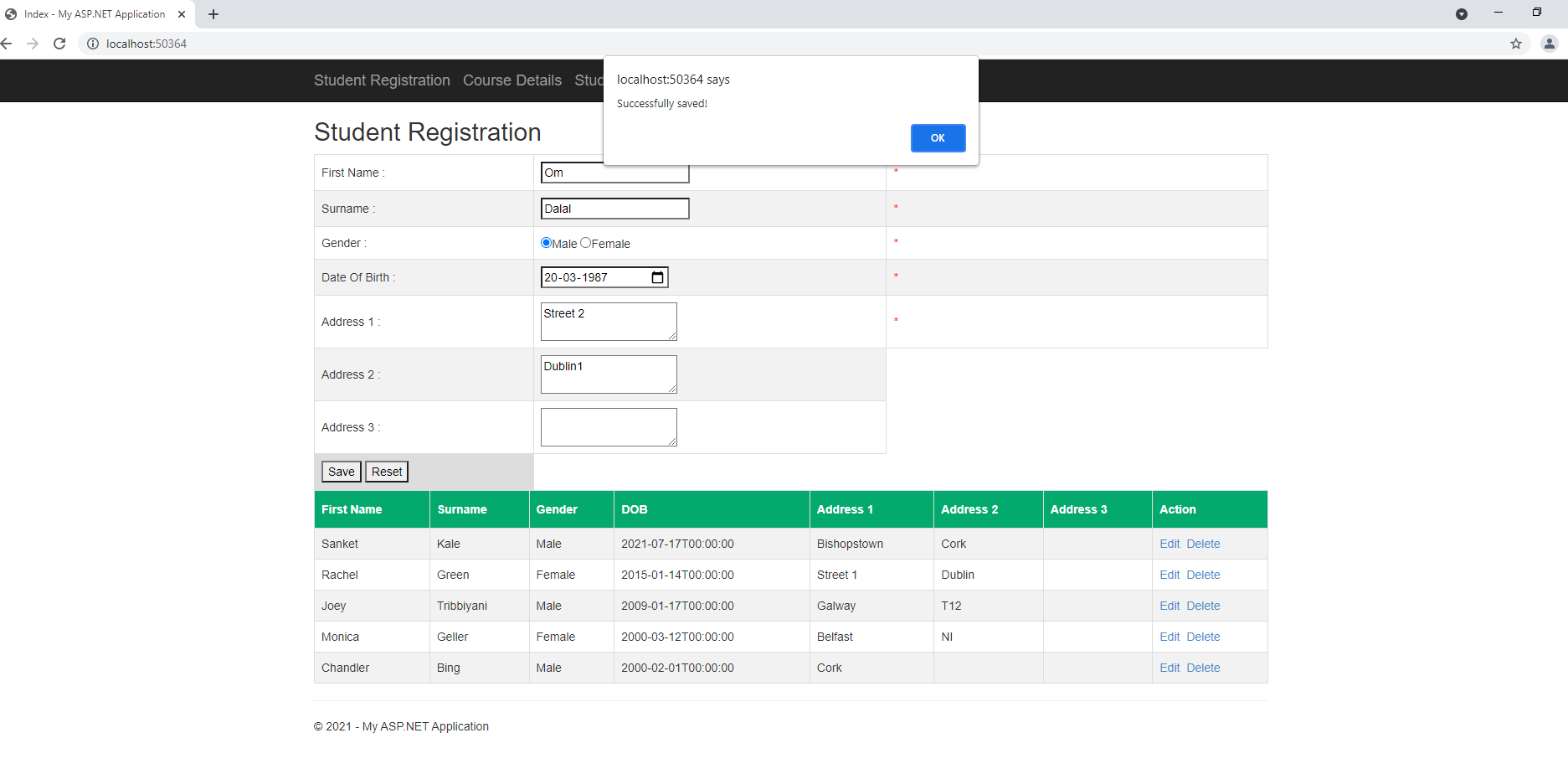


We can Add Update delete the student details. Check the validation for the mandatory fields.

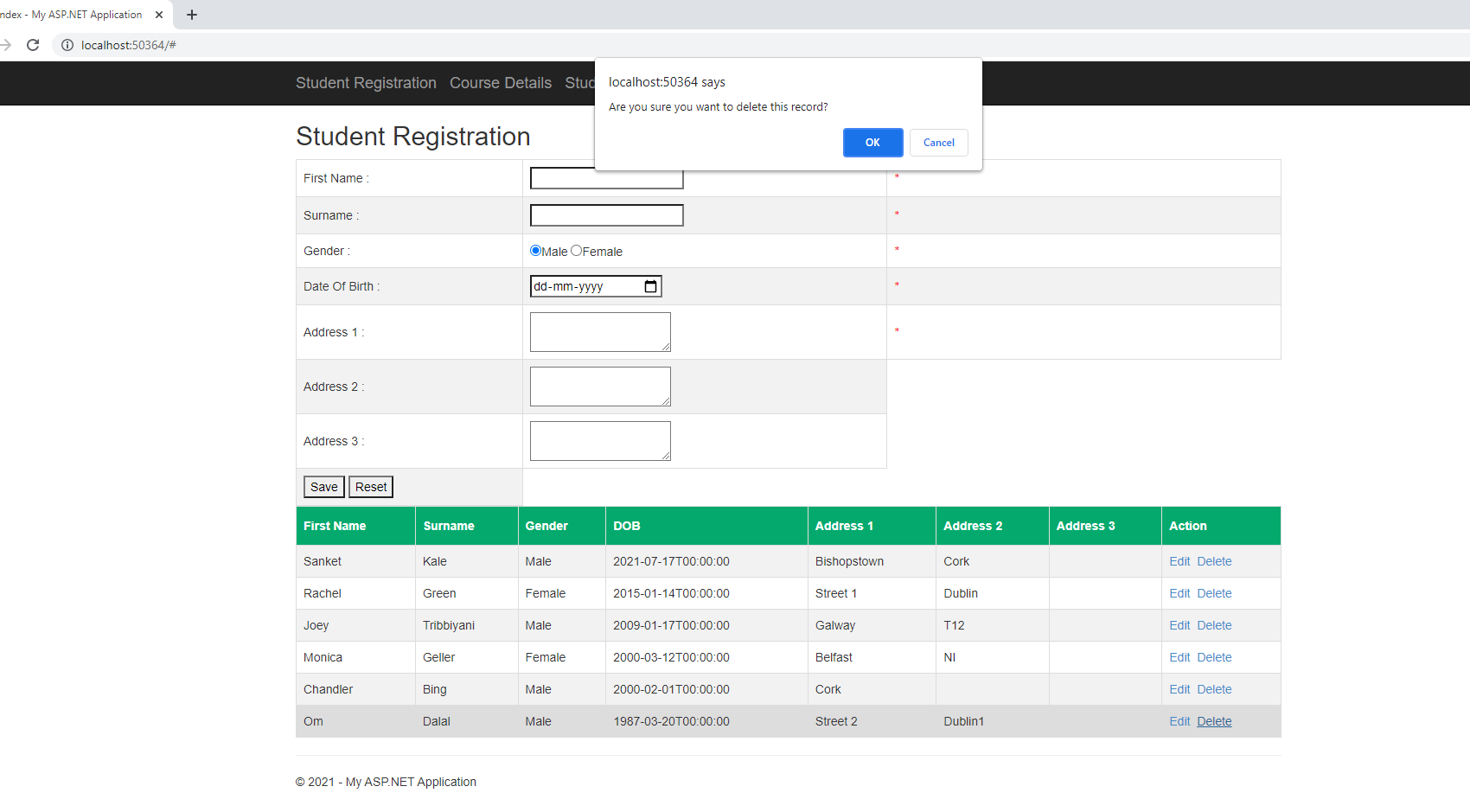
Om the top of menu bar we have following options

* Student Registration – To add update delete the student details
* Course Details – To add update and delete the Course details
* StudentCourse – To add Student and course mapping

Add Student

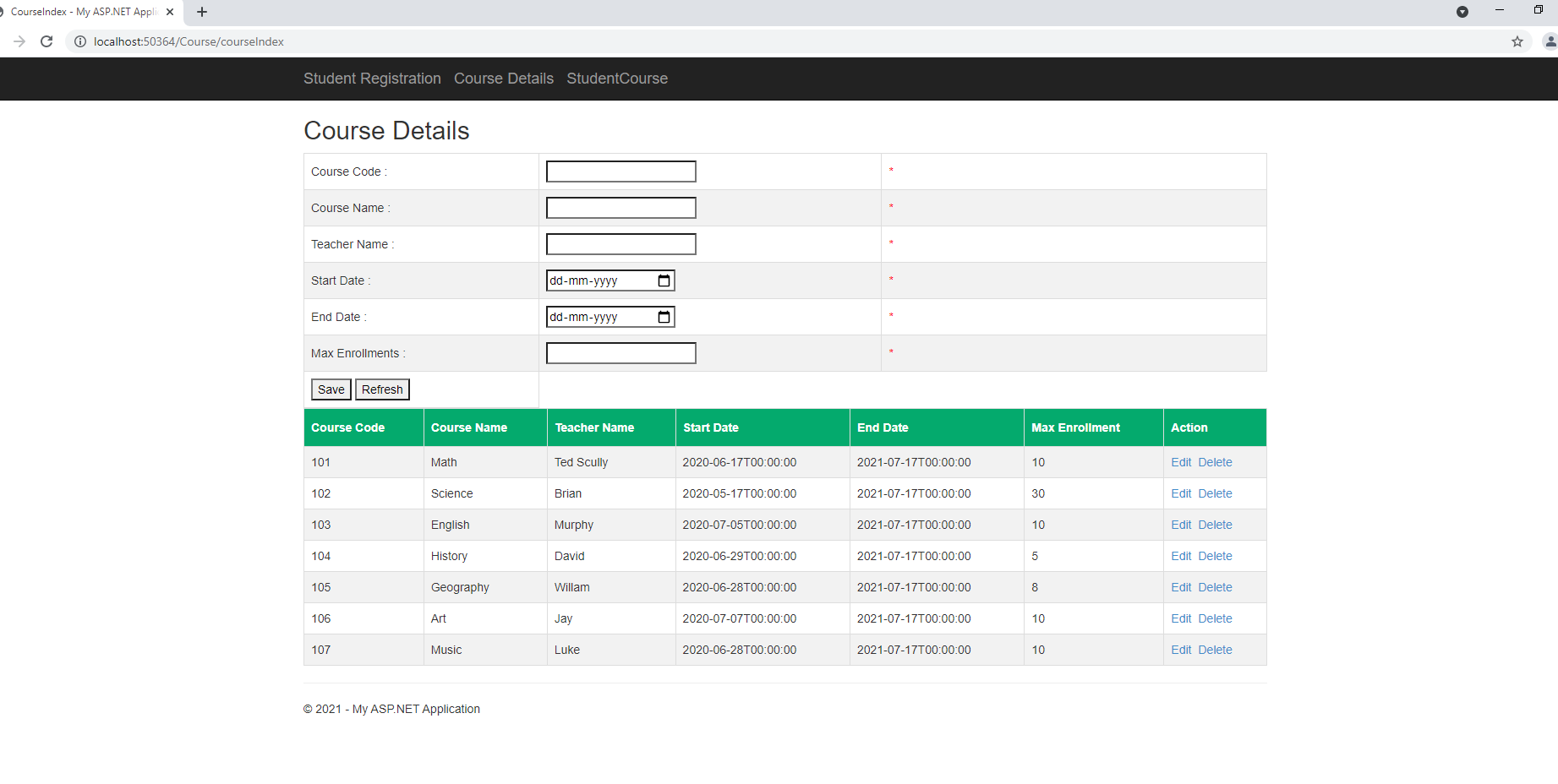


Delete data



 Course Details

* Controller – CourseController
* View – Course/Index.cshtml
* Business Layer – CourseModel.cs
* Database Layer – CourseDetails.cs

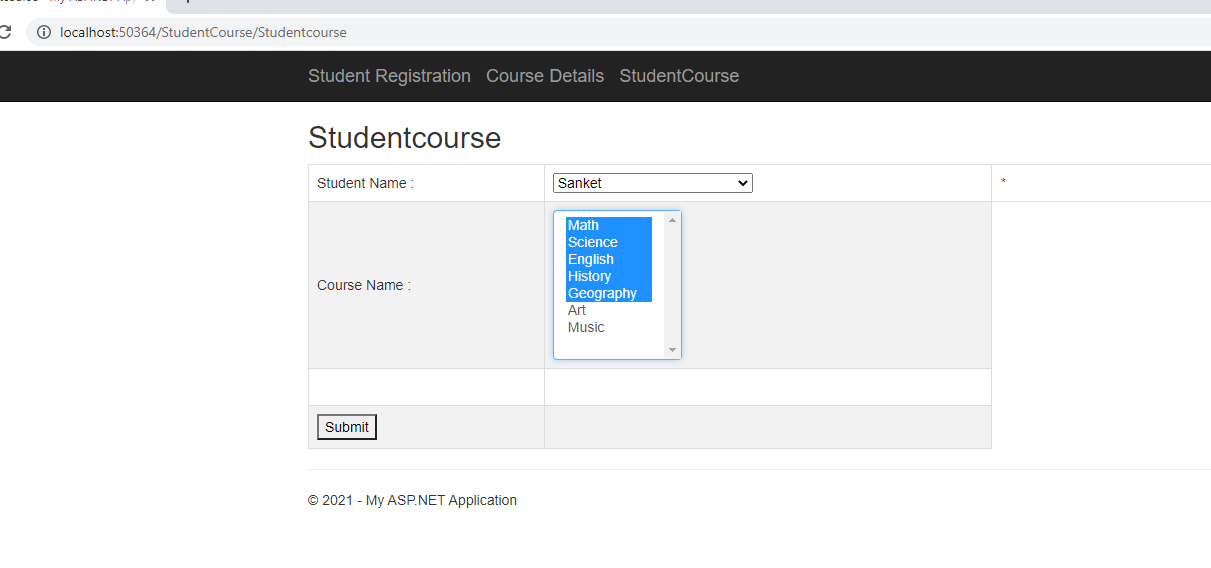


We can Add Update delete the course details and Max enrollment field added to check the total space in the course

Note: CourseCode is integer type

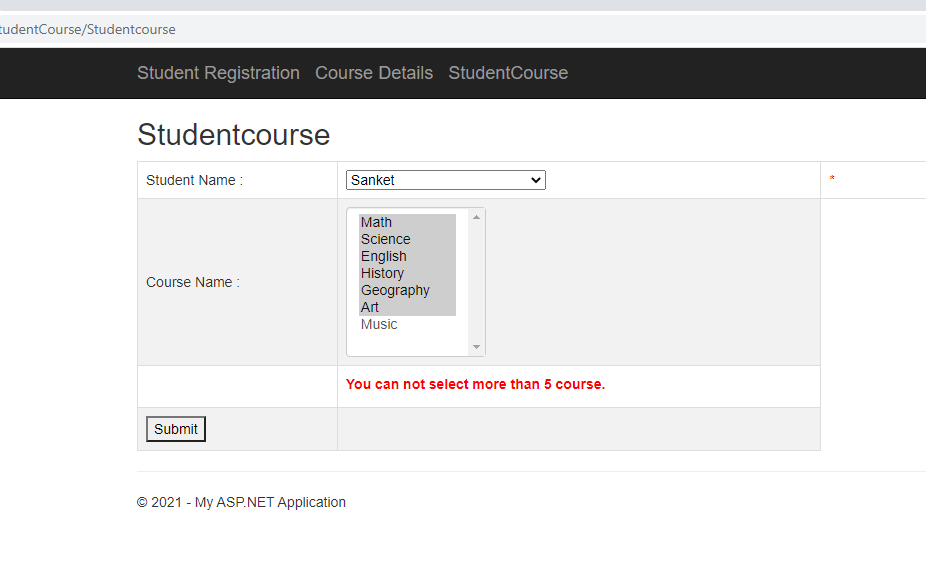
Student Course

* Controller – StudentCourseController
* View – StudentCourse/Index.cshtml
* BusinessLayer – StudentCourseModel.cs
* DatabaseLayer – StdCourse.cs

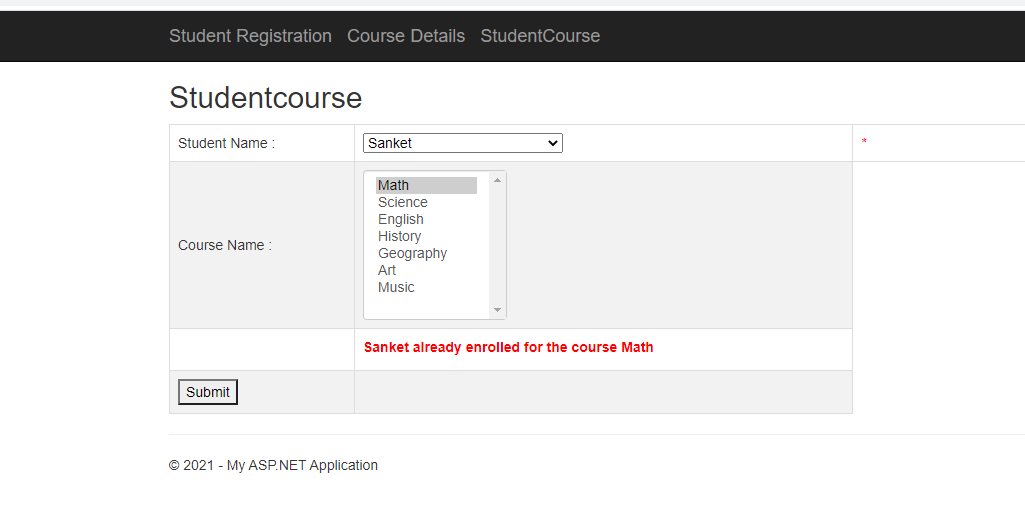


Validations

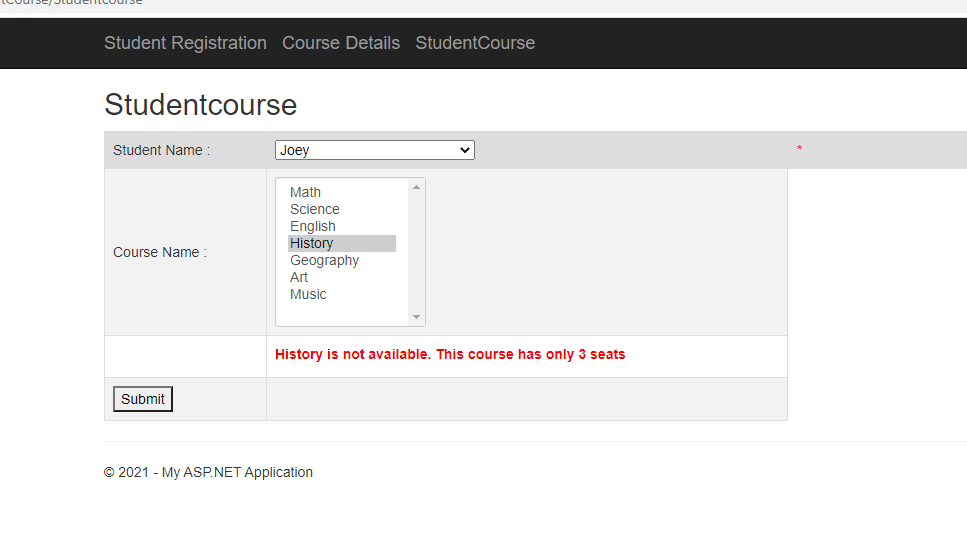
You can not add more than 5 courses to student



The following student already enrolled for the course



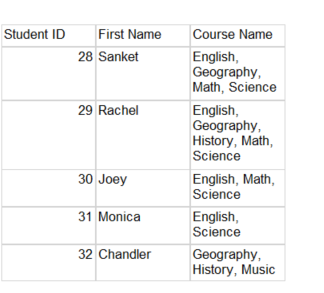
 The course history has only 3 seats, you cannot enroll the more student for that course



Task 2

SQL Report with store procedure

1. Show registered courses for each student



Create procedure [dbo].[sp\_showstudewisedata]

as

select distinct std.StudentID,std.FirstName,

STUFF((SELECT distinct ', ' + crs.CourseName

from tblStudentCourse sc join tblCourse crs on crs.CourseCode = sc.CourseCode

where sc.StudentId = std.[StudentId]

FOR XML PATH(''), TYPE

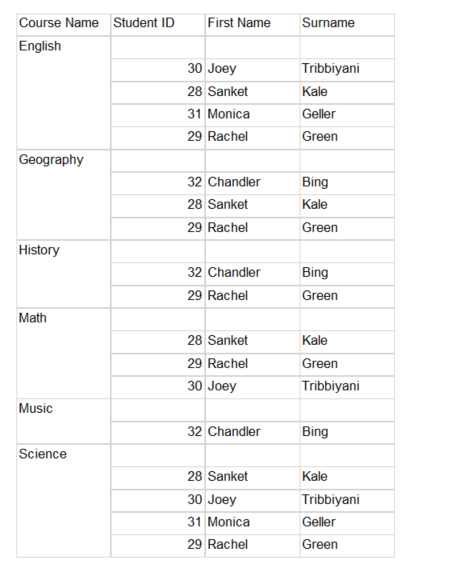
).value('.', 'NVARCHAR(MAX)')

,1,2,'') CourseName

from tblstudent std join tblStudentCourse sc

on sc.StudentId = std.[StudentId]

1. Show student list for each courses



Create procedure [dbo].[sp\_coursewise\_student]

as

select crs.CourseName, std.StudentID,std.FirstName,std.Surname

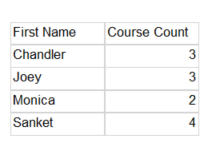
from tblCourse crs join tblStudentCourse sc

on crs.CourseCode = sc.CourseCode

join tblStudent std

on std.StudentId = sc.StudentId

1. Show how many students didn’t register max course



Create procedure [dbo].[sp\_student\_lessthanfive\_Course]

as

select std.FirstName,count(sc.[CourseCode])

from tblStudent std

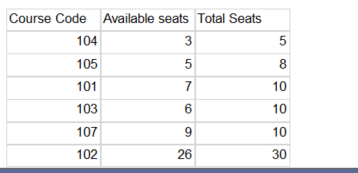
join tblStudentCourse sc

on std.StudentId = sc.StudentId

group by std.FirstName

having count(sc.[CourseCode]) < 5

1. Show how many courses still have the space



Create procedure [dbo].[sp\_availableseats]

as

select sc.CourseCode, cc.[MaxCourseCount] - Count(sc.[CourseCode]) as Available\_seats, cc.[MaxCourseCount] as Total\_Seats

from [dbo].[tblStudentCourse] sc

join [dbo].[tblCourseCount] cc

on sc.CourseCode = cc.CourseCode

group by sc.CourseCode,cc.[MaxCourseCount]