

Create an MVC Application to Manage Data for School Application

DESCRIPTION

Using ASP.NET MVC project to create a web based application, which will allow an admin to do CRUD operations on Student, Subject and Classe tables in the database

Background of the problem statement:

Rainbow School is creating software for school management. After creating the database, the first step is to populate the database using a web-based application.

You must use the following:

Visual Studio 2019 Community Edition

SQL Server 17 Express Edition

Following requirements should be met:

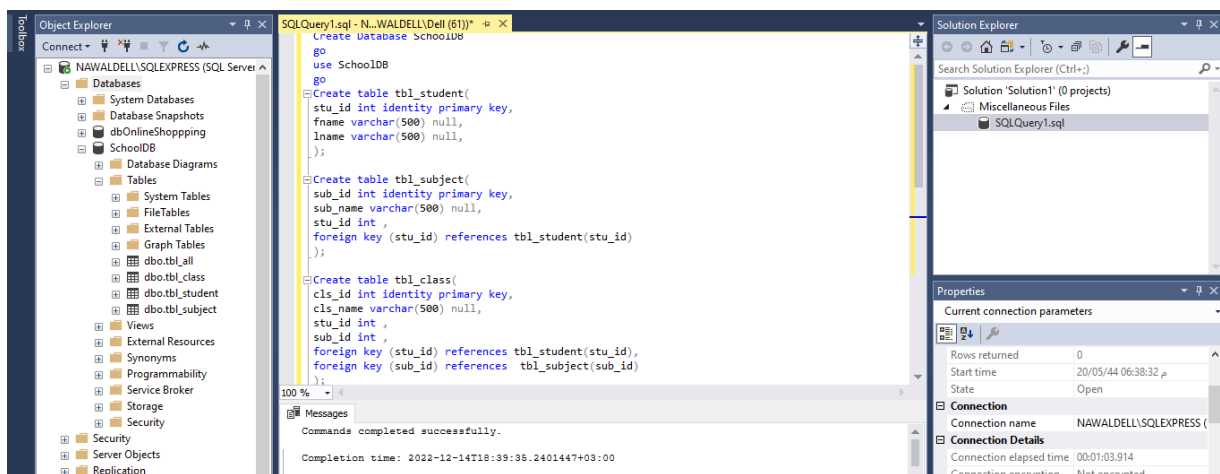
Some of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.

The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.

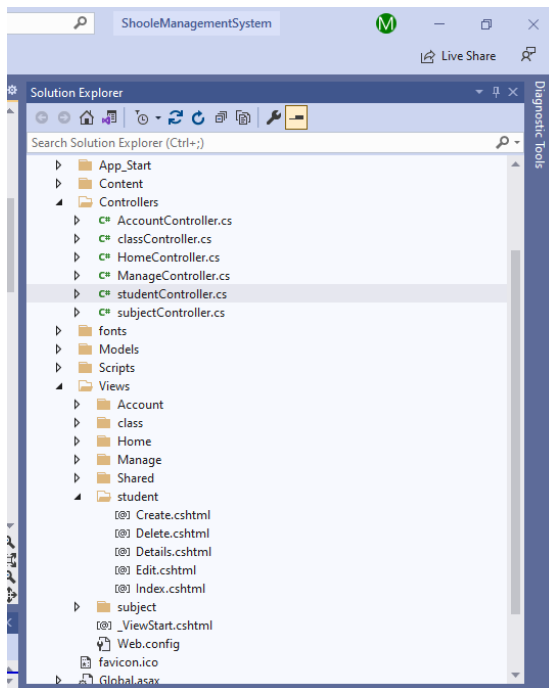
The step-by-step process involved in completing this task should be documented.

Source Code

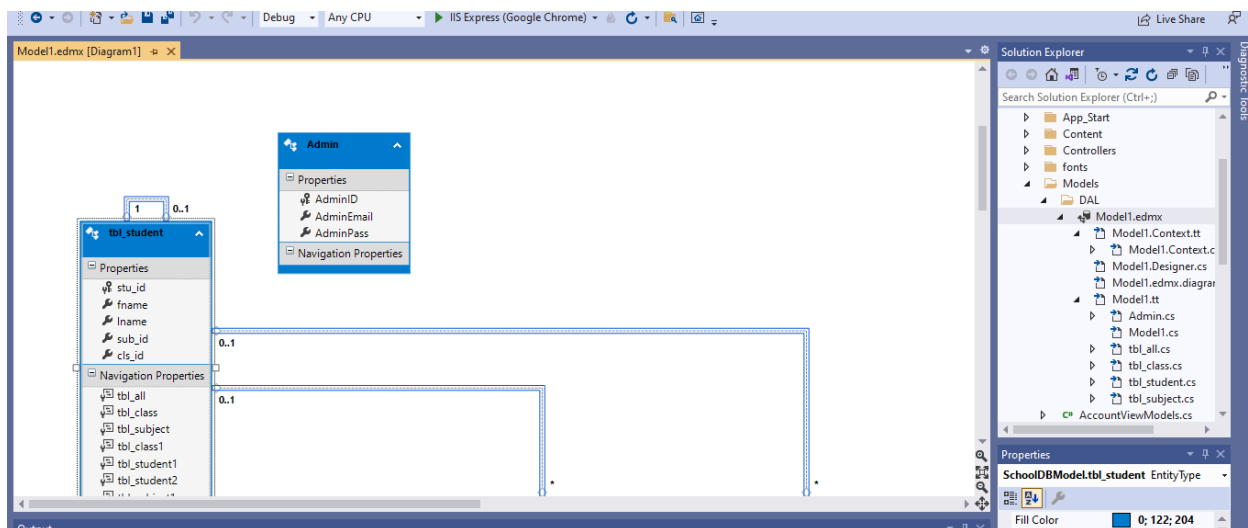
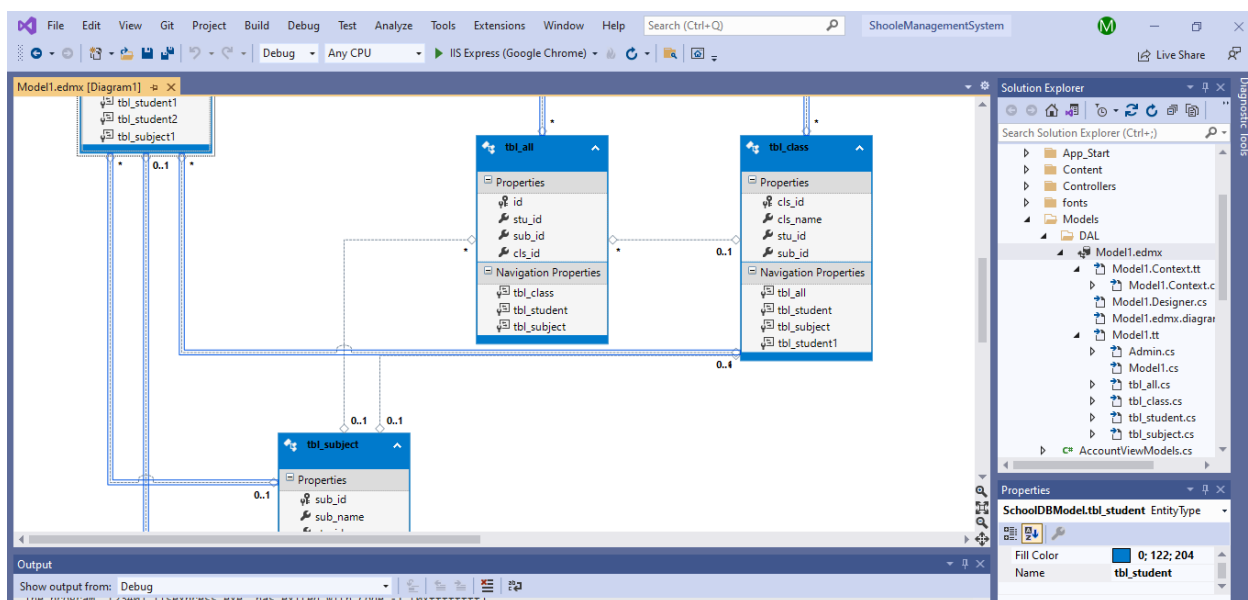
1- DB



2- The files of Project



3- DB diagram



4- Student CDUD – controller

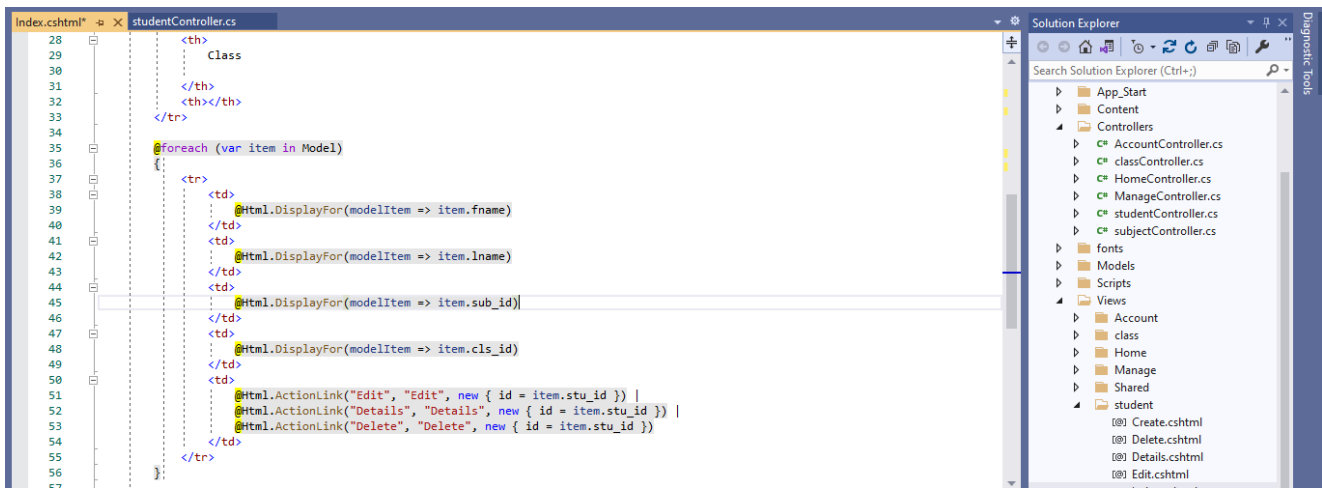
```
ShooleManagementSystem - ShooleManagementSystem.Controllers.studentController - Index()
10
11 namespace ShooleManagementSystem.Controllers
12 {
13     public class studentController : Controller
14     {
15         private SchoolDBEntities1 db = new SchoolDBEntities1();
16
17         // GET: student
18         public ActionResult Index()
19         {
20             return View(db.tbl_student.ToList());
21         }
22
23         // GET: student/Details/5
24         public ActionResult Details(int? id)
25         {
26             if (id == null)
27             {
28                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
29             }
30             tbl_student tbl_student = db.tbl_student.Find(id);
31             if (tbl_student == null)
32             {
33                 return HttpNotFound();
34             }
35             return View(tbl_student);
36         }
37     }
38 }
```

```
studentController.cs - Model1.edmx [Diagram1] - ShooleManagementSystem.Controllers.studentController - Index()
34 }
35 return View(tbl_student);
36 }
37
38 // GET: student/Create
39 public ActionResult Create()
40 {
41     return View();
42 }
43
44 // POST: student/Create
45 // To protect from overposting attacks, enable the specific properties you want to bind to, for
46 // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
47 [HttpPost]
48 [ValidateAntiForgeryToken]
49 public ActionResult Create([Bind(Include = "stu_id,fname,lname,sub_id,cls_id")] tbl_student tbl_student)
50 {
51     if (ModelState.IsValid)
52     {
53         db.tbl_student.Add(tbl_student);
54         db.SaveChanges();
55         return RedirectToAction("Index");
56     }
57     return View(tbl_student);
58 }
59
60 // GET: student/Edit/5
61
```

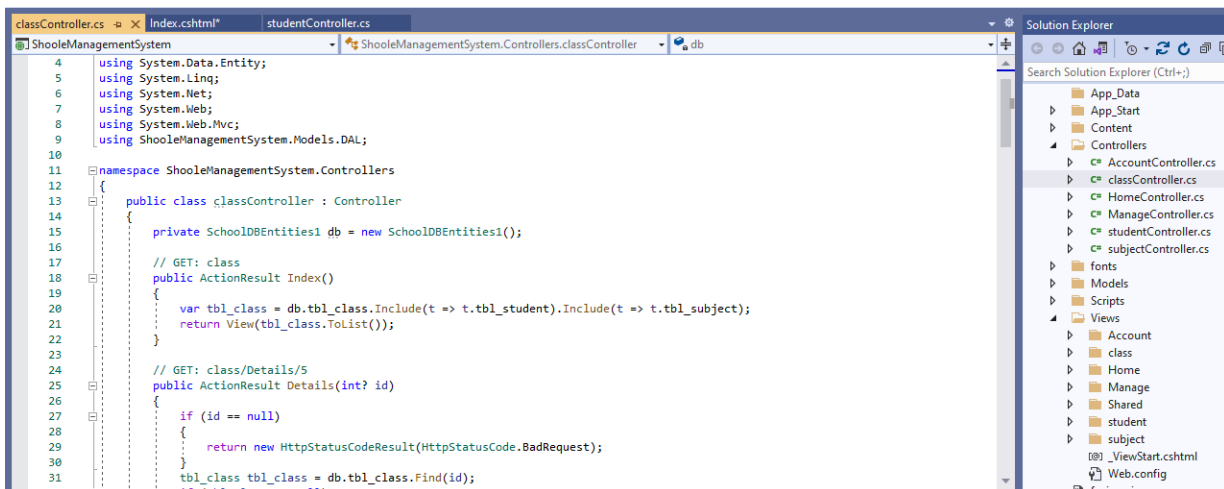
```
studentController.cs - ShooleManagementSystem.Controllers.studentController - db
61 // GET: student/Edit/5
62 public ActionResult Edit(int? id)
63 {
64     if (id == null)
65     {
66         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
67     }
68     tbl_student tbl_student = db.tbl_student.Find(id);
69     if (tbl_student == null)
70     {
71         return HttpNotFound();
72     }
73     return View(tbl_student);
74 }
75
76 // POST: student/Edit/5
77 // To protect from overposting attacks, enable the specific properties you want to bind to, for
78 // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
79 [HttpPost]
80 [ValidateAntiForgeryToken]
81 public ActionResult Edit([Bind(Include = "stu_id,fname,lname,sub_id,cls_id")] tbl_student tbl_student)
82 {
83     if (ModelState.IsValid)
84     {
85         db.Entry(tbl_student).State = EntityState.Modified;
86         db.SaveChanges();
87         return RedirectToAction("Index");
88     }
89 }
```

```
studentController.cs - ShooleManagementSystem.Controllers.studentController - db
91
92 // GET: student/Delete/5
93 public ActionResult Delete(int? id)
94 {
95     if (id == null)
96     {
97         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
98     }
99     tbl_student tbl_student = db.tbl_student.Find(id);
100     if (tbl_student == null)
101     {
102         return HttpNotFound();
103     }
104     return View(tbl_student);
105 }
106
107 // POST: student/Delete/5
108 [HttpPost, ActionName("Delete")]
109 [ValidateAntiForgeryToken]
110 public ActionResult DeleteConfirmed(int id)
111 {
112     tbl_student tbl_student = db.tbl_student.Find(id);
113     db.tbl_student.Remove(tbl_student);
114     db.SaveChanges();
115     return RedirectToAction("Index");
116 }
117
118 protected override void Dispose(bool disposing)
```

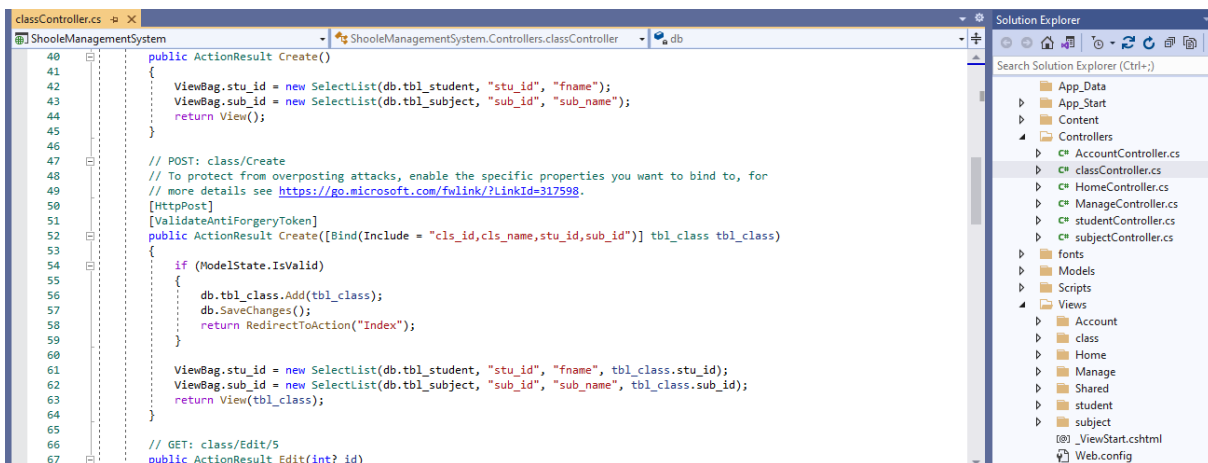
Student – index page



5- Class – CRUD Controller



Class – create view



Class – Edit View

```

classController.cs
ShooleManagementSystem ShooleManagementSystem.Controllers.classController db
67 public ActionResult Edit(int? id)
68 {
69     if (id == null)
70     {
71         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
72     }
73     tbl_class tbl_class = db.tbl_class.Find(id);
74     if (tbl_class == null)
75     {
76         return HttpNotFound();
77     }
78     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_class.stu_id);
79     ViewBag.sub_id = new SelectList(db.tbl_subject, "sub_id", "sub_name", tbl_class.sub_id);
80     return View(tbl_class);
81 }
82
83 // POST: class/Edit/5
84 // To protect from overposting attacks, enable the specific properties you want to bind to, for
85 // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
86 [HttpPost]
87 [ValidateAntiForgeryToken]
88 public ActionResult Edit([Bind(Include = "cls_id,cls_name,stu_id,sub_id")] tbl_class tbl_class)
89 {
90     if (ModelState.IsValid)
91     {
92         db.Entry(tbl_class).State = EntityState.Modified;
93         db.SaveChanges();
94         return RedirectToAction("Index");
95     }
96 }

```

Class – delete view

```

classController.cs ShooleManagementSystem ShooleManagementSystem.Controllers.classController Create(tbl_class tbl_class)
85 [ValidateAntiForgeryToken]
86 public ActionResult Edit([Bind(Include = "cls_id,cls_name,stu_id,sub_id")] tbl_class tbl_class)
87 {
88     if (ModelState.IsValid)
89     {
90         db.Entry(tbl_class).State = EntityState.Modified;
91         db.SaveChanges();
92         return RedirectToAction("Index");
93     }
94     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_class.stu_id);
95     ViewBag.sub_id = new SelectList(db.tbl_subject, "sub_id", "sub_name", tbl_class.sub_id);
96     return View(tbl_class);
97 }
98
99 // GET: class/Delete/5
100 public ActionResult Delete(int? id)
101 {
102     if (id == null)
103     {
104         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
105     }
106     tbl_class tbl_class = db.tbl_class.Find(id);
107     if (tbl_class == null)
108     {
109         return HttpNotFound();
110     }
111     return View(tbl_class);
112 }
113
114 // POST: class/Delete/5
115 [HttpPost, ActionName("Delete")]
116 [ValidateAntiForgeryToken]
117 public ActionResult DeleteConfirmed(int id)
118 {
119     tbl_class tbl_class = db.tbl_class.Find(id);
120     db.tbl_class.Remove(tbl_class);
121     db.SaveChanges();
122     return RedirectToAction("Index");
123 }
124
125 protected override void Dispose(bool disposing)
126 {
127     if (disposing)
128     {
129         db.Dispose();
130     }
131     base.Dispose(disposing);
132 }
133
134 }
135
enum System.Net.HttpStatusCode
Contains the values of status codes defined for HTTP.

```

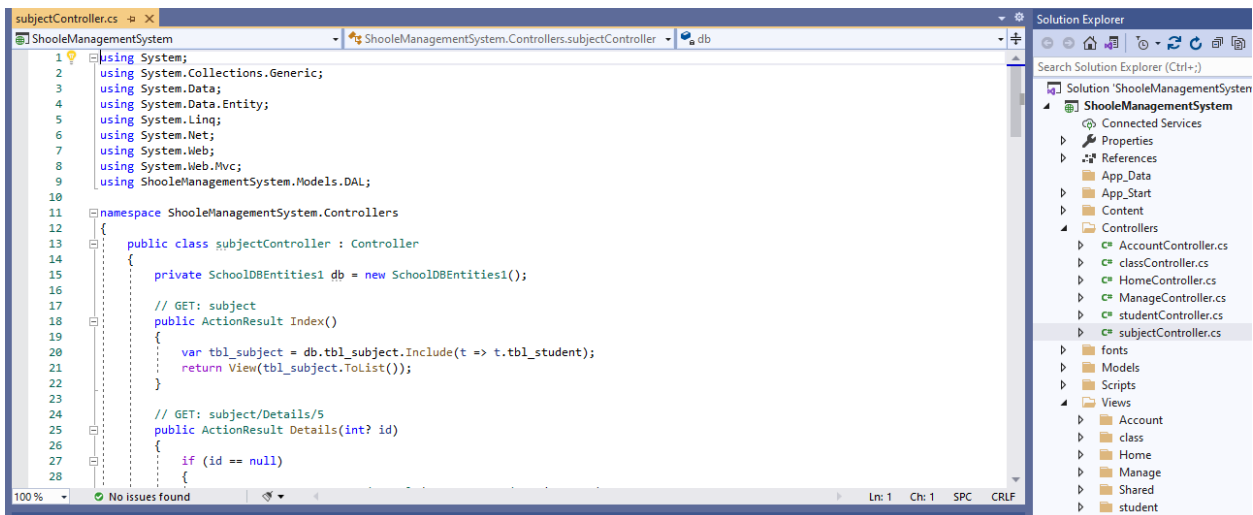
Class – index page

```

Index.cshtml classController.cs
16 <p>
17     @Html.ActionLink("Create New", "Create")
18 </p>
19 <table class="table">
20 <tr>
21 <th>
22     Class Name
23 </th>
24 <th>
25     Subject Name
26 </th>
27 <th>
28     </th>
29 </tr>
30 </table>
31
32 @foreach (var item in Model) {
33 <tr>
34 <td>
35     @Html.DisplayFor(modelItem => item.cls_name)
36 </td>
37 <td>
38     @Html.DisplayFor(modelItem => item.tbl_subject.sub_name)
39 </td>
40 <td>
41     @Html.ActionLink("Edit", "Edit", new { id=item.cls_id }) |
42     @Html.ActionLink("Details", "Details", new { id=item.cls_id }) |
43     @Html.ActionLink("Delete", "Delete", new { id=item.cls_id })
44 </td>
45 </tr>
46 }

```

6- Subject – CRUD operations -- Controller



```
1 using System;
2 using System.Collections.Generic;
3 using System.Data;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Net;
7 using System.Web;
8 using System.Web.Mvc;
9 using ShoolManagementSystem.Models.DAL;

10 namespace ShoolManagementSystem.Controllers
11 {
12     public class subjectController : Controller
13     {
14         private SchoolDBEntities1 db = new SchoolDBEntities1();

15         // GET: subject
16         public ActionResult Index()
17         {
18             var tbl_subject = db.tbl_subject.Include(t => t.tbl_student);
19             return View(tbl_subject.ToList());
20         }

21         // GET: subject/Details/5
22         public ActionResult Details(int? id)
23         {
24             if (id == null)
25             {
26                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
27             }
28             tbl_subject tbl_subject = db.tbl_subject.Find(id);
29             if (tbl_subject == null)
30             {
31                 return HttpNotFound();
32             }
33             return View(tbl_subject);
34         }

35         // GET: subject/Create
36         public ActionResult Create()
37         {
38             ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname");
39             return View();
40         }

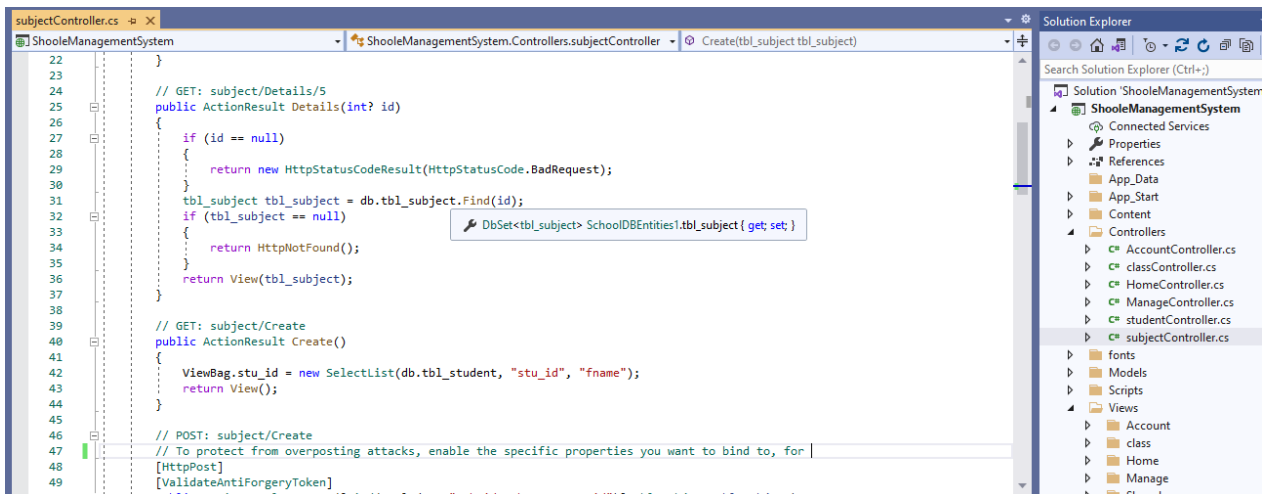
41         // POST: subject/Create
42         // To protect from overposting attacks, enable the specific properties you want to bind to, for
43         // example: { firstName, lastName, email, password, confirmPassword}
44         [HttpPost]
45         [ValidateAntiForgeryToken]
46         public ActionResult Create([Bind(Include = "sub_id,sub_name,stu_id")] tbl_subject tbl_subject)
47         {
48             if (ModelState.IsValid)
49             {
50                 db.tbl_subject.Add(tbl_subject);
51                 db.SaveChanges();
52                 return RedirectToAction("Index");
53             }
54             ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);
55             return View(tbl_subject);
56         }

57         // GET: subject/Edit/5
58         public ActionResult Edit(int? id)
59         {
60             if (id == null)
61             {
62                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
63             }
64             tbl_subject tbl_subject = db.tbl_subject.Find(id);
65             if (tbl_subject == null)
66             {
67                 return HttpNotFound();
68             }
69             return View(tbl_subject);
70         }

71         // PUT: subject/Edit/5
72         // To protect from overposting attacks, enable the specific properties you want to bind to, for
73         // example: { firstName, lastName, email, password, confirmPassword}
74         [HttpPost]
75         [ValidateAntiForgeryToken]
76         public ActionResult Edit([Bind(Include = "sub_id,sub_name,stu_id")] tbl_subject tbl_subject)
77         {
78             if (ModelState.IsValid)
79             {
80                 db.Entry(tbl_subject).State = EntityState.Modified;
81                 db.SaveChanges();
82                 return RedirectToAction("Index");
83             }
84             ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);
85             return View(tbl_subject);
86         }

87         // DELETE: subject/Edit/5
88         public ActionResult Delete(int? id)
89         {
90             if (id == null)
91             {
92                 return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
93             }
94             tbl_subject tbl_subject = db.tbl_subject.Find(id);
95             if (tbl_subject == null)
96             {
97                 return HttpNotFound();
98             }
99             db.tbl_subject.Remove(tbl_subject);
100             db.SaveChanges();
101             return RedirectToAction("Index");
102         }
103     }
104 }
```

Subject – details & Create views



```
22 }
23
24 // GET: subject/Details/5
25 public ActionResult Details(int? id)
26 {
27     if (id == null)
28     {
29         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
30     }
31     tbl_subject tbl_subject = db.tbl_subject.Find(id);
32     if (tbl_subject == null)
33     {
34         return HttpNotFound();
35     }
36     return View(tbl_subject);
37 }
38
39 // GET: subject/Create
40 public ActionResult Create()
41 {
42     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname");
43     return View();
44 }
45
46 // POST: subject/Create
47 // To protect from overposting attacks, enable the specific properties you want to bind to, for
48 // example: { firstName, lastName, email, password, confirmPassword}
49 [HttpPost]
50 [ValidateAntiForgeryToken]
51 public ActionResult Create([Bind(Include = "sub_id,sub_name,stu_id")] tbl_subject tbl_subject)
52 {
53     if (ModelState.IsValid)
54     {
55         db.tbl_subject.Add(tbl_subject);
56         db.SaveChanges();
57         return RedirectToAction("Index");
58     }
59     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);
60     return View(tbl_subject);
61 }
62
63 // GET: subject/Edit/5
64 public ActionResult Edit(int? id)
65 {
66     if (id == null)
67     {
68         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
69     }
70     tbl_subject tbl_subject = db.tbl_subject.Find(id);
71     if (tbl_subject == null)
72     {
73         return HttpNotFound();
74     }
75     return View(tbl_subject);
76 }
77
78 // PUT: subject/Edit/5
79 // To protect from overposting attacks, enable the specific properties you want to bind to, for
80 // example: { firstName, lastName, email, password, confirmPassword}
81 [HttpPost]
82 [ValidateAntiForgeryToken]
83 public ActionResult Edit([Bind(Include = "sub_id,sub_name,stu_id")] tbl_subject tbl_subject)
84 {
85     if (ModelState.IsValid)
86     {
87         db.Entry(tbl_subject).State = EntityState.Modified;
88         db.SaveChanges();
89         return RedirectToAction("Index");
90     }
91     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);
92     return View(tbl_subject);
93 }
94
95 // DELETE: subject/Edit/5
96 public ActionResult Delete(int? id)
97 {
98     if (id == null)
99     {
100         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
101     }
102     tbl_subject tbl_subject = db.tbl_subject.Find(id);
103     if (tbl_subject == null)
104     {
105         return HttpNotFound();
106     }
107     db.tbl_subject.Remove(tbl_subject);
108     db.SaveChanges();
109     return RedirectToAction("Index");
110 }
111 }
```

Subject – Edit view

```

subjectController.cs
ShooleManagementSystem
ShooleManagementSystem.Controllers.subjectController
Edit(tbl_subject tbl_subject)

64 public ActionResult Edit(int? id)
65 {
66     if (id == null)
67     {
68         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
69     }
70     tbl_subject tbl_subject = db.tbl_subject.Find(id);
71     if (tbl_subject == null)
72     {
73         return HttpNotFound();
74     }
75     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);
76     return View(tbl_subject);
77 }
78
79 // POST: subject/Edit/5
80 // To protect from overposting attacks, enable the specific properties you want to bind to, for
81 [HttpPost]
82 [ValidateAntiForgeryToken]
83 public ActionResult Edit([Bind(Include = "sub_id,sub_name,stu_id")] tbl_subject tbl_subject)
84 {
85     if (ModelState.IsValid)
86     {
87         db.Entry(tbl_subject).State = EntityState.Modified;
88         db.SaveChanges();
89         return RedirectToAction("Index");
90     }
91     ViewBag.stu_id = new SelectList(db.tbl_student, "stu_id", "fname", tbl_subject.stu_id);

```

Subject – delete view

```

subjectController.cs
ShooleManagementSystem
ShooleManagementSystem.Controllers.subjectController
Edit(tbl_subject tbl_subject)

94
95 // GET: subject/Delete/5
96 public ActionResult Delete(int? id)
97 {
98     if (id == null)
99     {
100         return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
101     }
102     tbl_subject tbl_subject = db.tbl_subject.Find(id);
103     if (tbl_subject == null)
104     {
105         return HttpNotFound();
106     }
107     return View(tbl_subject);
108 }
109
110 // POST: subject/Delete/5
111 [HttpPost, ActionName("Delete")]
112 [ValidateAntiForgeryToken]
113 public ActionResult DeleteConfirmed(int id)
114 {
115     tbl_subject tbl_subject = db.tbl_subject.Find(id);
116     db.tbl_subject.Remove(tbl_subject);
117     db.SaveChanges();
118     return RedirectToAction("Index");
119 }
120
121 protected override void Dispose(bool disposing)
122 {

```

```

subjectController.cs
ShooleManagementSystem
ShooleManagementSystem.Controllers.subjectController
Edit(tbl_subject tbl_subject)

109
110 // POST: subject/Delete/5
111 [HttpPost, ActionName("Delete")]
112 [ValidateAntiForgeryToken]
113 public ActionResult DeleteConfirmed(int id)
114 {
115     tbl_subject tbl_subject = db.tbl_subject.Find(id);
116     db.tbl_subject.Remove(tbl_subject);
117     db.SaveChanges();
118     return RedirectToAction("Index");
119 }
120
121 protected override void Dispose(bool disposing)
122 {
123     if (disposing)
124     {
125         db.Dispose();
126     }
127     base.Dispose(disposing);
128 }
129
130
131

```

Subject – index page

```

Index.cshtml
subjectController.cs

13 </div>
14 </div>
15
16 <p>
17     @Html.ActionLink("Create New", "Create")
18 </p>
19 <table class="table">
20 <tr>
21 <th>
22     Subject Name
23 </th>
24 <th>Student Name</th>
25 </tr>
26
27 <tr>
28 <td>
29     @foreach (var item in Model) {
30 <tr>
31 <td>
32     @Html.DisplayFor(modelItem => item.sub_name)
33 </td>
34 <td>
35     @Html.DisplayFor(modelItem => item.tbl_student.fname)
36 </td>
37 <td>
38     @Html.ActionLink("Edit", "Edit", new { id=item.sub_id }) |
39     @Html.ActionLink("Details", "Details", new { id=item.sub_id }) |
40     @Html.ActionLink("Delete", "Delete", new { id=item.sub_id })
41 </td>
42 </tr>

```