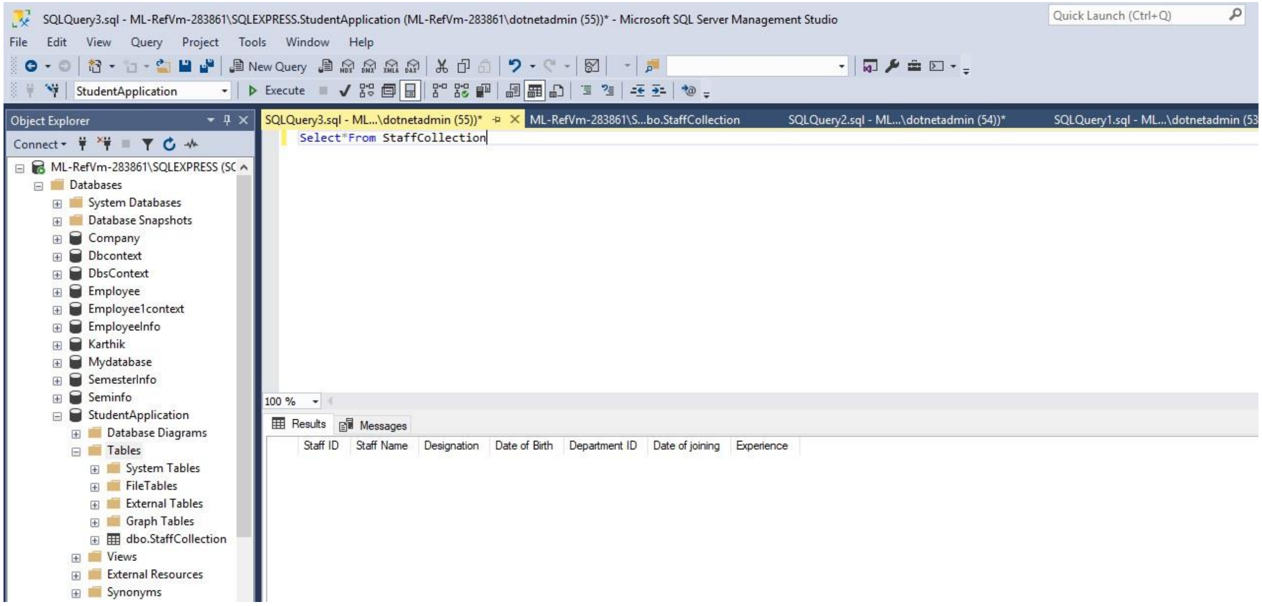
Milestone Assessment 5

Name : Priyansu Jena

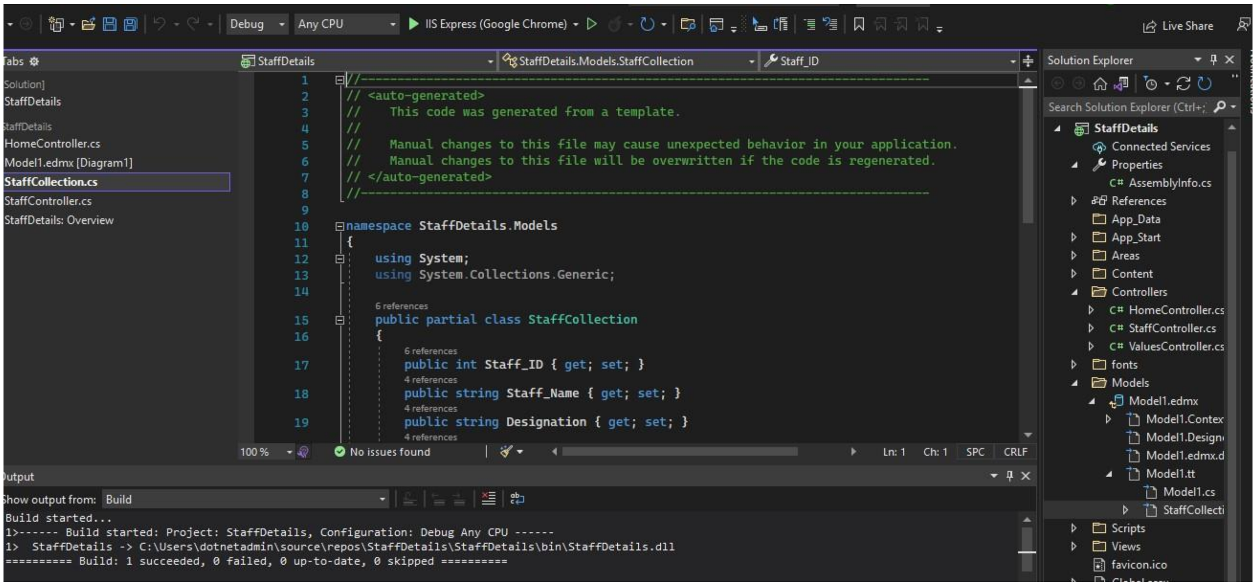
Mid : M1082626


SQLQuery3.sql - ML-RefVm-283861\SQLEXPRESS.StudentApplication (ML-RefVm-283861\dotnetadmin (55))* - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
StudentApplication
Execute
Object Explorer
Connect
ML-RefVm-283861\SQLEXPRESS (SC
Databases
System Databases
Database Snapshots
Company
Dbcontext
DbsContext
Employee
Employee1context
EmployeeInfo
Karthik
Mydatabase
SemesterInfo
Seminfo
StudentApplication
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.StaffCollection
Views
External Resources
Synonyms
SQLQuery3.sql - ML...\dotnetadmin (55))*
ML-RefVm-283861\S...bo.StaffCollection
SQLQuery2.sql - ML...\dotnetadmin (54))*
SQLQuery1.sql - ML...\dotnetadmin (53
100 %
Results
Messages

```sql
Select*From StaffCollection
```

Staff ID	Staff Name	Designation	Date of Birth	Department ID	Date of joining	Experience

Controller Model



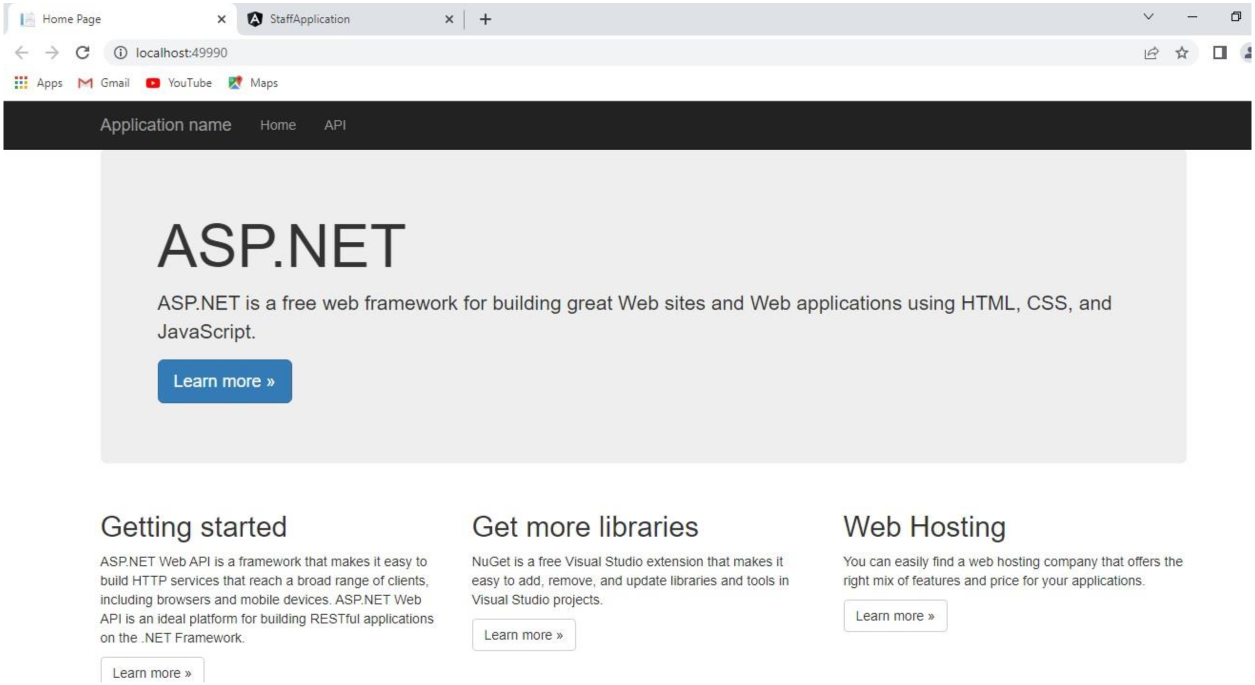
```csharp
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated from a template.
//
//     Manual changes to this file may cause unexpected behavior in your application.
//     Manual changes to this file will be overwritten if the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace StaffDetails.Models
{
    using System;
    using System.Collections.Generic;

    6 references
    public partial class StaffCollection
    {
        6 references
        public int Staff_ID { get; set; }
        4 references
        public string Staff_Name { get; set; }
        4 references
        public string Designation { get; set; }
        4 references
```

```
Build started...
1>------ Build started: Project: StaffDetails, Configuration: Debug Any CPU ------
1>  StaffDetails -> C:\Users\dotnetadmin\source\repos\StaffDetails\StaffDetails\bin\StaffDetails.dll
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Controller Code

using System;
using System.Collections.Generic; using
System.Linq; using System.Net; using

```
System.Net.Http; using System.Web.Http;
using StaffDetails.Models;


namespace StaffDetails.Controllers
{
    public class StaffController : ApiController
    {
        StudentApplicationEntities EN = new StudentApplicationEntities();
        [Route("StudentInsert")]
        [HttpPost]
        public object StudentInsert(StaffCollection objVM)
        {            try
{
                if (objVM.Staff_ID == 0)
                {
                    var objlm = new StaffCollection();                    objlm.Staff_Name = objVM.Staff_Name;
objlm.Designation = objVM.Designation;                    objlm.Date_of_Birth = objVM.Date_of_Birth;
objlm.Department_ID = objVM.Department_ID;                    objlm.Date_of_joining =
objVM.Date_of_joining;                    objlm.Experience = objVM.Experience;


                    EN.StaffCollections.Add(objlm);
                    EN.SaveChanges();                    return new
                    { Status = "Success", Message = "SuccessFully Saved." };
                }                else
{
                    var objlm = EN.StaffCollections.Where(s => s.Staff_ID ==
objVM.Staff_ID).ToList<StaffCollection>().FirstOrDefault();                    if (objlm.Staff_ID > 0)
                    {
                        objlm.Staff_Name = objVM.Staff_Name;                        objlm.Designation = objVM.Designation;
objlm.Date_of_Birth = objVM.Date_of_Birth;                        objlm.Department_ID = objVM.Department_ID;
objlm.Date_of_joining = objVM.Date_of_joining;                        objlm.Department_ID =
objVM.Department_ID;

                        EN.SaveChanges();                        return new
                        { Status = "Success", Message = "SuccessFully Update." };                    }                    return new
                    { Status = "Error", Message = "Invalid." };
                }            }
            catch (Exception ex)
            {
                return new
                { Status = "Error", Message = ex.Message.ToString() };
            }
        }
        [Route("GetStudentData")]
        [HttpGet]
        public object GetStudentData()
        {
            var obj = from u in EN.StaffCollections                    select u;
return obj;
        }
        [Route("GetStudentById")]
        [HttpGet]
        public object GetStudentById(int Id)
        {
            return EN.StaffCollections.Where(s => s.Staff_ID ==
Id).ToList<StaffCollection>().FirstOrDefault();
        }
        [Route("DeleteStudent")]
        [HttpGet]
        public object  DeleteStudent(int Id)
        {       try
{
                var objlm = EN.StaffCollections.Where(s => s.Staff_ID ==
Id).ToList<StaffCollection>().FirstOrDefault();
                EN.StaffCollections.Remove(objlm);
                EN.SaveChanges();                 return new
                { Status = "Success", Message = "SuccessFully Delete." };
            }
            catch (Exception ex)
            {
                return new
                { Status = "Error", Message = ex.Message.ToString() };
            }
        }
    }
}
```



ASP.NET

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

Learn more »

Getting started

ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework.

Learn more »

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

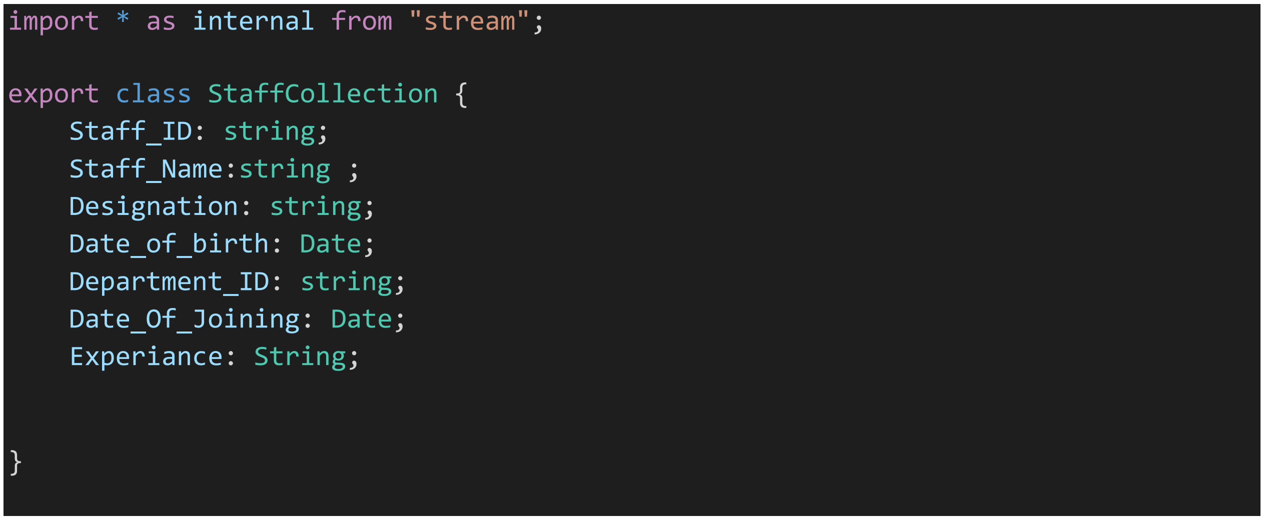
Learn more »

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

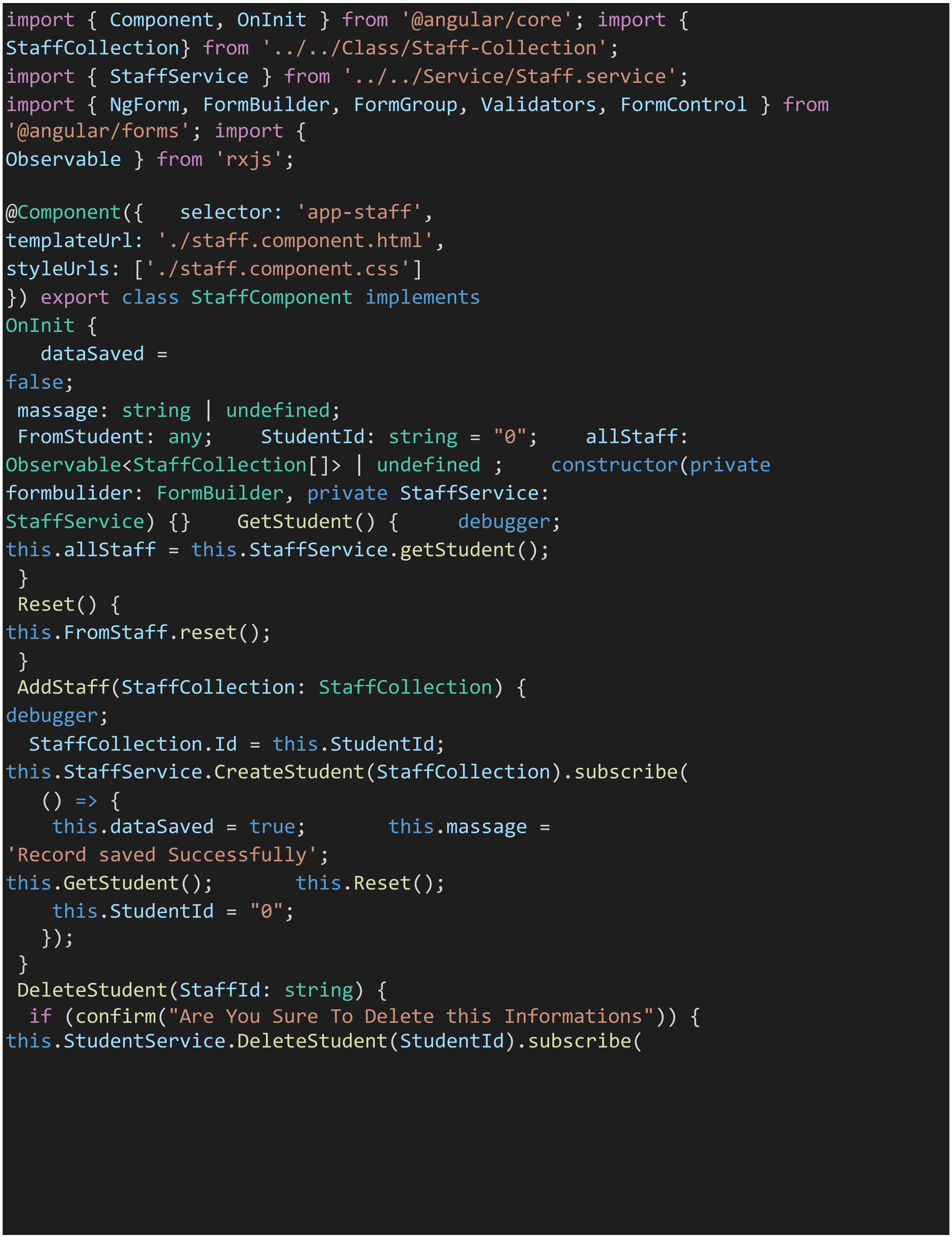
Learn more »

StaffCollection.ts



```
import * as internal from "stream";

export class StaffCollection {
    Staff_ID: string;
    Staff_Name:string ;
    Designation: string;
    Date_of_birth: Date;
    Department_ID: string;
    Date_Of_Joining: Date;
    Experiance: String;


}
```

StaffComponent.ts



```
import { Component, OnInit } from '@angular/core'; import {
StaffCollection} from '../../Class/Staff-Collection';
import { StaffService } from '../../Service/Staff.service';
import { NgForm, FormBuilder, FormGroup, Validators, FormControl } from
'@angular/forms'; import {
Observable } from 'rxjs';

@Component({   selector: 'app-staff',
templateUrl: './staff.component.html',
styleUrls: ['./staff.component.css']
}) export class StaffComponent implements
OnInit {
   dataSaved =
false;
 massage: string | undefined;
 FromStudent: any;    StudentId: string = "0";    allStaff:
Observable<StaffCollection[]> | undefined ;    constructor(private
formbulider: FormBuilder, private StaffService:
StaffService) {}    GetStudent() {     debugger;
this.allStaff = this.StaffService.getStudent();
 }
 Reset() {
this.FromStaff.reset();
 }
 AddStaff(StaffCollection: StaffCollection) {
debugger;
  StaffCollection.Id = this.StudentId;
this.StaffService.CreateStudent(StaffCollection).subscribe(
   () => {
    this.dataSaved = true;       this.massage =
'Record saved Successfully';
this.GetStudent();       this.Reset();
    this.StudentId = "0";
   });
 }
 DeleteStudent(StaffId: string) {
  if (confirm("Are You Sure To Delete this Informations")) {
this.StudentService.DeleteStudent(StudentId).subscribe(
    () => {        this.dataSaved = true;
this.massage = "Deleted Successfully";
this.GetStudent();
    }
   );
  }
 }
 StudentEdit(StudentId: string) {     debugger;
this.StudentService.getStudentById(StudentId).subscribe((Response: { Staff_ID:
string; Staff_Name: any; Designation: any; Date_of_Birth: any; Department_ID:
any;  }) => {
         this.dataSaved = false;      debugger;      this.StudentId =
Response.Staff_ID;
this.FromStaff.controls['StaffName'].setValue(Response.Staff_Name);
this.FromStaff.controls['Designation'].setValue(Response.Designation);
this.FromStaff.controls['Date of Birth'].setValue(Response.Date_of_Birth);
this.FromStaff.controls['Experience'].setValue(Response.Department_ID);
  });    }
ngOnInit(): void {
  this.FromStudent      =      this.formbulider.group({
StaffId: ['', [Validators.required]],
   StaffName: ['', [Validators.required]],
   Designation: ['', [Validators.required]],
   Date of Joining: ['', [Validators.required]],
   Experience: ['', [Validators.required]],
  });
  this.GetStaff();
 }
```

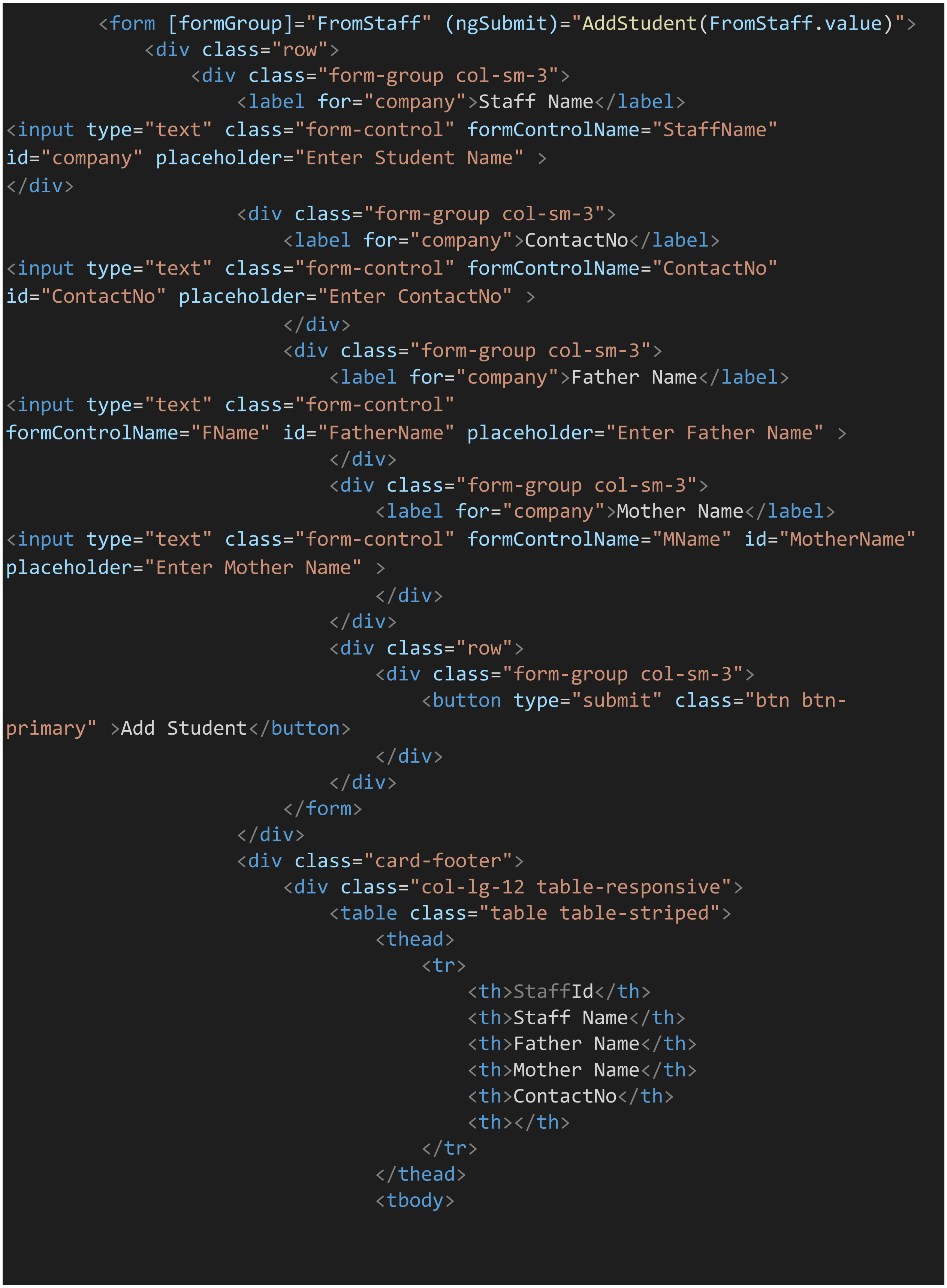
StaffService.ts

```
import { Injectable } from '@angular/core';     import
{HttpClient} from '@angular/common/http';     import
{HttpHeaders} from '@angular/common/http';     import {
Observable } from 'rxjs';     import { StaffCollection }
from '../Class/StaffCollection';
```

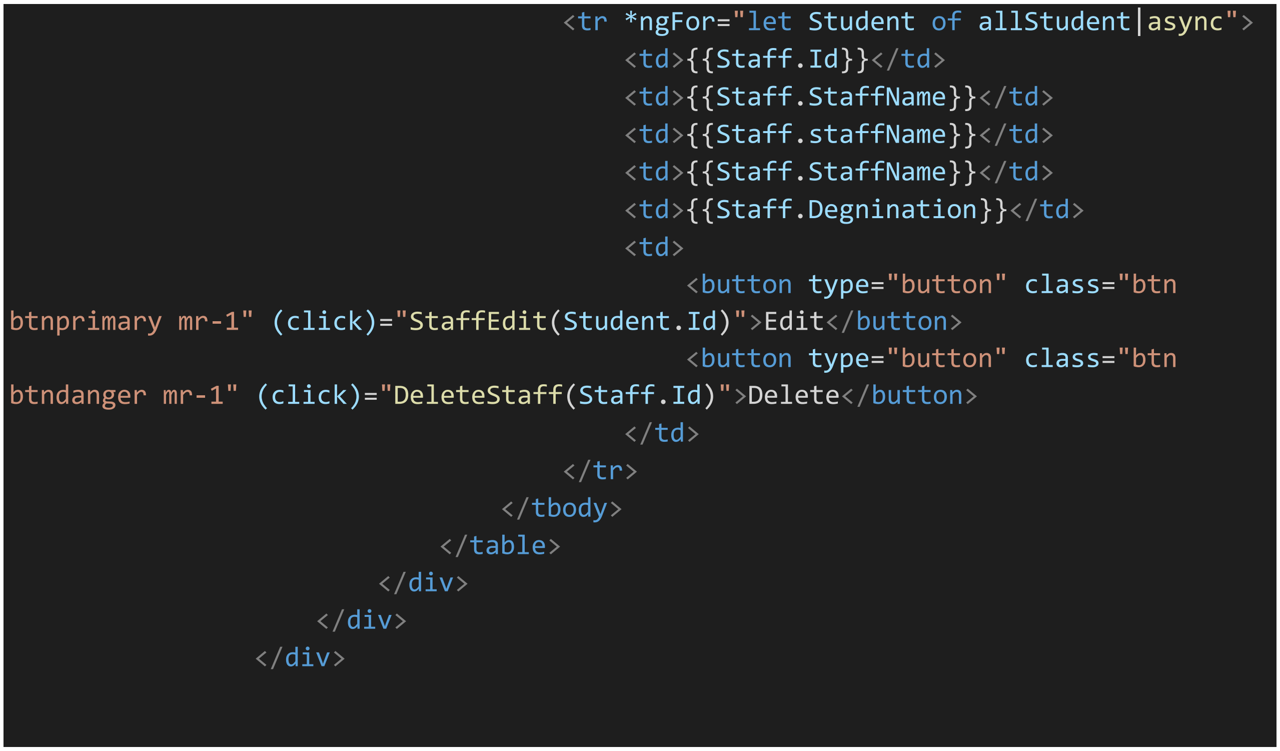
```
@Injectable({
providedIn: 'root'
}) export class
StaffService {

  Url = 'http://localhost:54996/Api';
constructor(private http:HttpClient) { }
getStaff():Observable<StaffCollection[]>
  {          return
this.http.get<StaffCollection[]>(this.Url +
'/student/GetStaffData');
  }
  CreateStaff(OutletVM:StaffCollection):Observable<StaffCollection[]>
  {        const httpOptions = { headers: new HttpHeaders({
'Content-Type':
'application/json' }) };
    return this.http.post<StaffCollection[]>(this.Url + '/student/StaffInsert/',
OutletVM, httpOptions)
  }
  DeleteStaff(StaffId:string):Observable<number>
  {          return
this.http.get<number>(this.Url +
'/student/DeleteStaff?Id='+StaffId);
  }        getStudentById(StaffId: string): Observable<StaffCollection> {
return this.http.get<StaffCollection>(this.Url + '/student/GetStaffById?Id='
+ Staff_Id);
  }
}

}
```

StaffComponent.Html

```
<p>staff works!</p>
<div class="card">
    <div class="card-header" style="text-align:center">
        <b>WEL COME TO STUDENT CURD OPERATION</b>
    </div>
    <div class="card-body">
```



```
        <form [formGroup]="FromStaff" (ngSubmit)="AddStudent(FromStaff.value)">
            <div class="row">
                <div class="form-group col-sm-3">
                    <label for="company">Staff Name</label>
<input type="text" class="form-control" formControlName="StaffName"
id="company" placeholder="Enter Student Name" >
</div>
                    <div class="form-group col-sm-3">
                        <label for="company">ContactNo</label>
<input type="text" class="form-control" formControlName="ContactNo"
id="ContactNo" placeholder="Enter ContactNo" >
                        </div>
                        <div class="form-group col-sm-3">
                            <label for="company">Father Name</label>
<input type="text" class="form-control"
formControlName="FName" id="FatherName" placeholder="Enter Father Name" >
                            </div>
                            <div class="form-group col-sm-3">
                                <label for="company">Mother Name</label>
<input type="text" class="form-control" formControlName="MName" id="MotherName"
placeholder="Enter Mother Name" >
                                </div>
                            </div>
                            <div class="row">
                                <div class="form-group col-sm-3">
                                    <button type="submit" class="btn btn-
primary" >Add Student</button>
                                </div>
                            </div>
                        </form>
                    </div>
                    <div class="card-footer">
                        <div class="col-lg-12 table-responsive">
                            <table class="table table-striped">
                                <thead>
                                    <tr>
                                        <th>StaffId</th>
                                        <th>Staff Name</th>
                                        <th>Father Name</th>
                                        <th>Mother Name</th>
                                        <th>ContactNo</th>
                                        <th></th>
                                    </tr>
                                </thead>
                                <tbody>
```



```
                                    <tr *ngFor="let Student of allStudent|async">
                                        <td>{{Staff.Id}}</td>
                                        <td>{{Staff.StaffName}}</td>
                                        <td>{{Staff.staffName}}</td>
                                        <td>{{Staff.StaffName}}</td>
                                        <td>{{Staff.Degnination}}</td>
                                        <td>
                                            <button type="button" class="btn
btnprimary mr-1" (click)="StaffEdit(Student.Id)">Edit</button>
                                            <button type="button" class="btn
btndanger mr-1" (click)="DeleteStaff(Staff.Id)">Delete</button>
                                        </td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
```