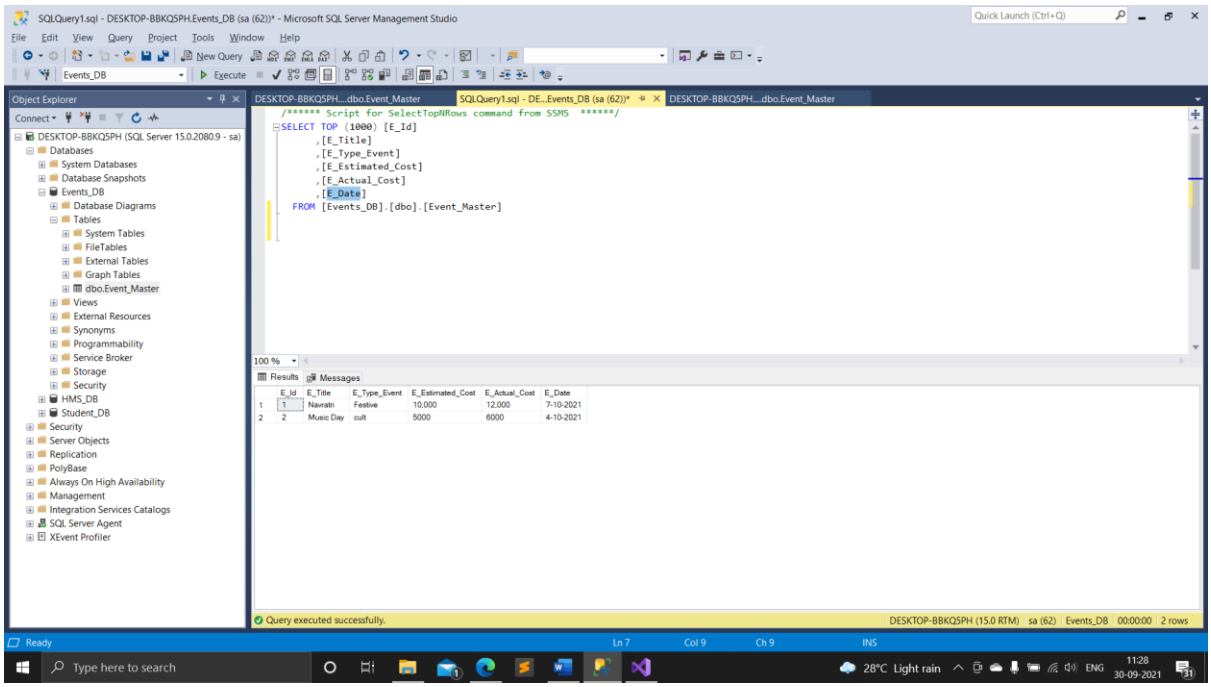
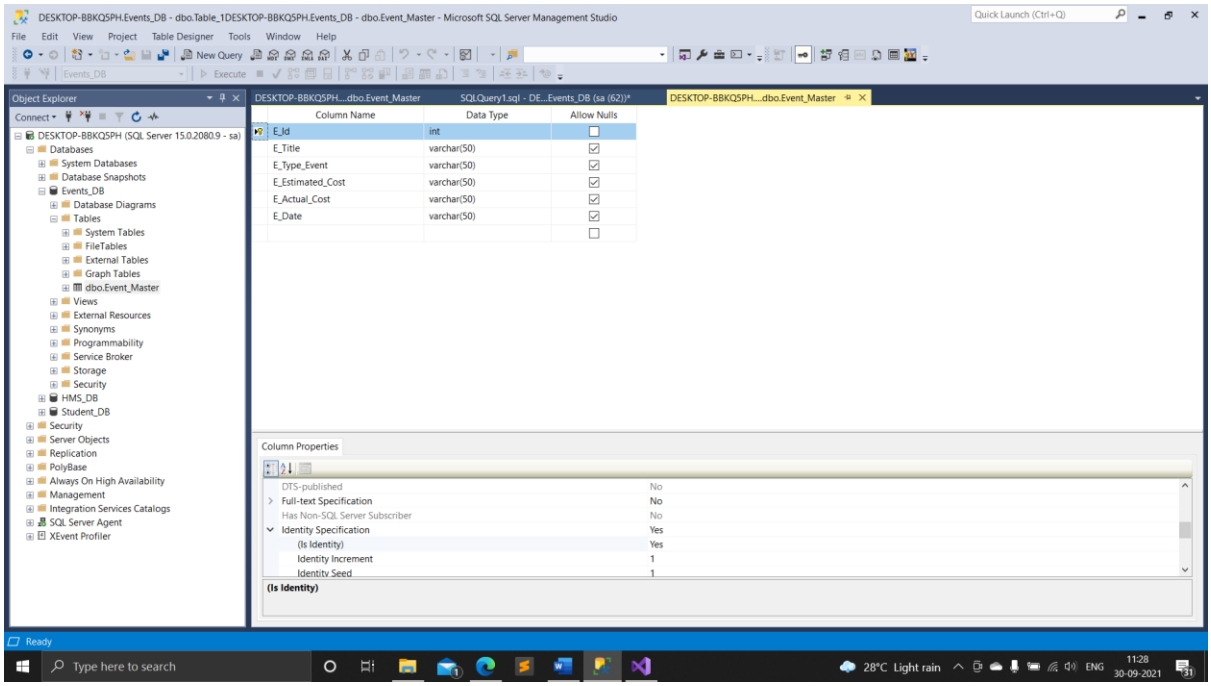
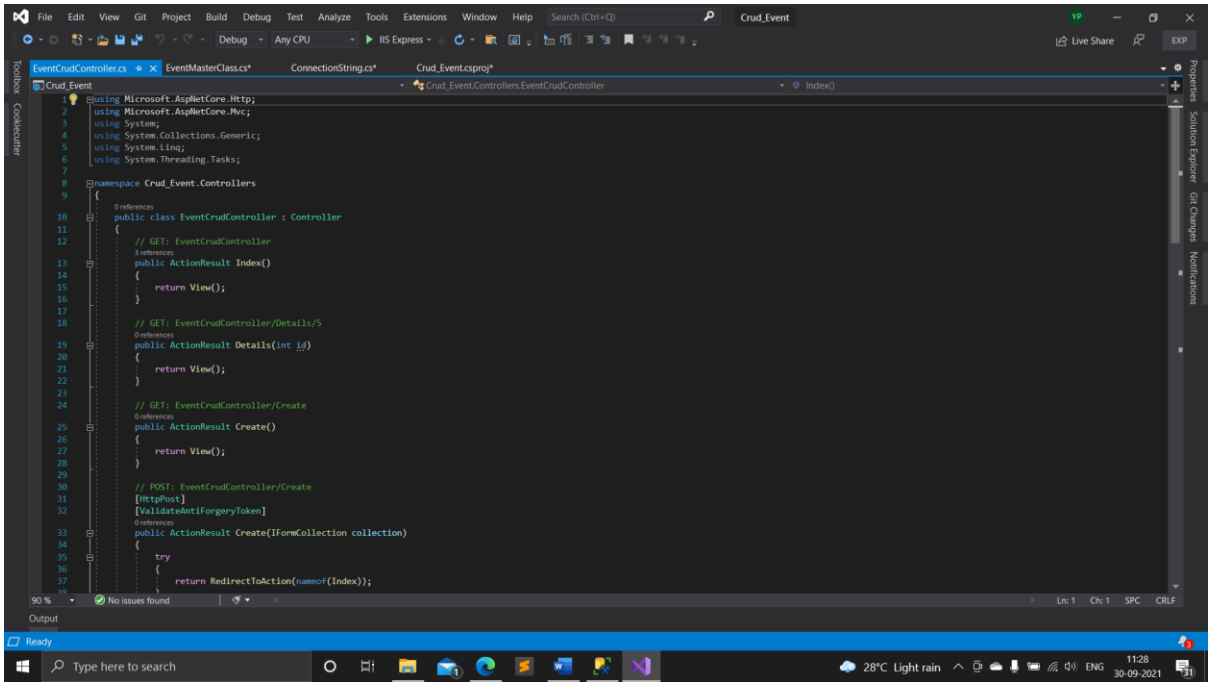
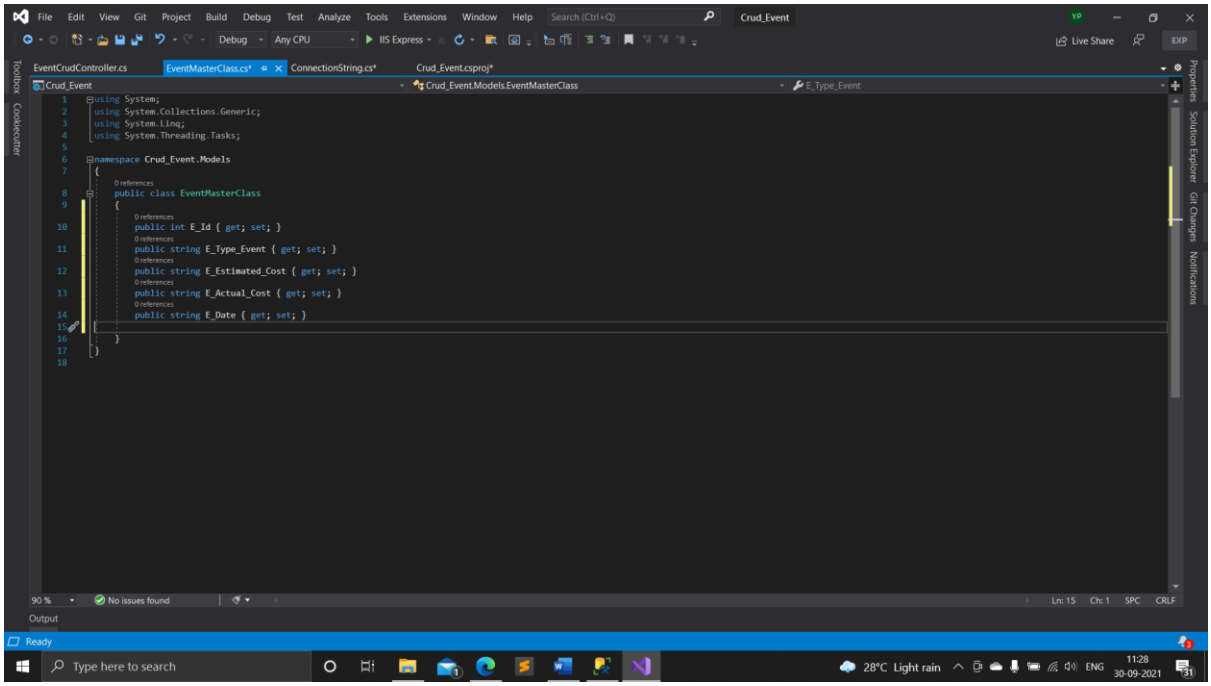
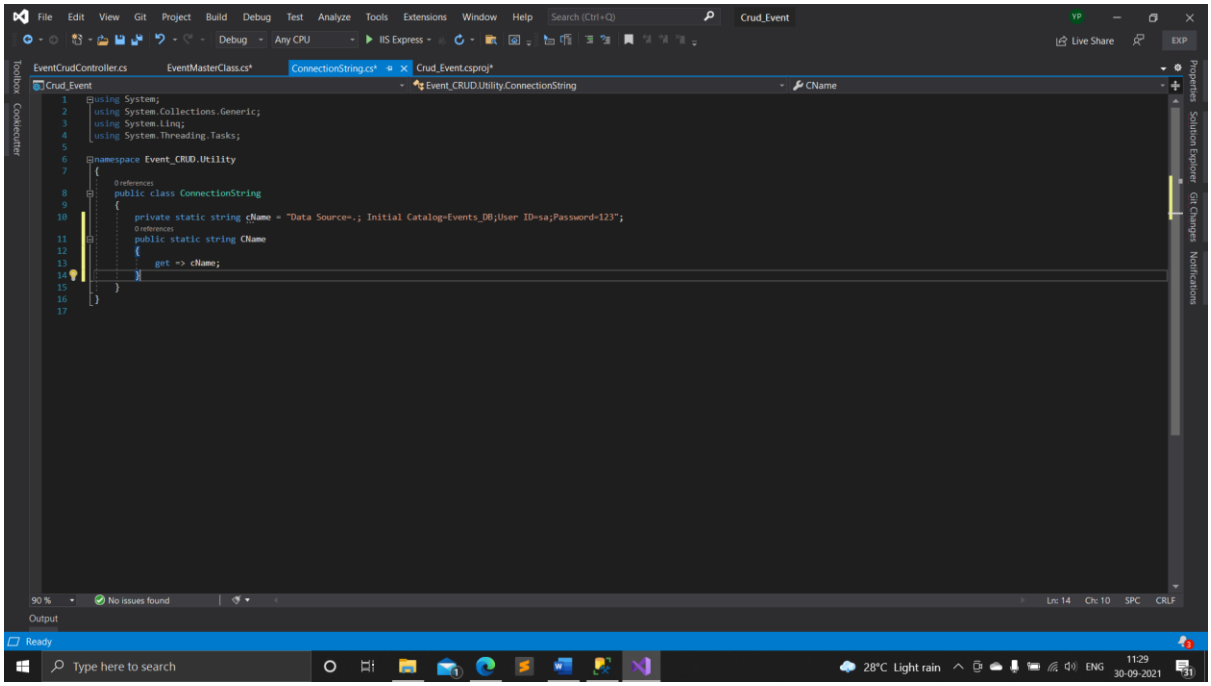
Practical Exam (205)






File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Crud_Event
Debug
Any CPU
IIS Express
Live Share
EXP
EventCrudController.cs
EventMasterClass.cs*
ConnectionString.cs*
Crud_Event.csproj*
Crud_Event
Crud_Event.Controllers.EventCrudController
Index()
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace Crud_Event.Controllers
{
0 references
public class EventCrudController : Controller
{
// GET: EventCrudController
3 references
public ActionResult Index()
{
return View();
}
// GET: EventCrudController/Details/5
0 references
public ActionResult Details(int id)
{
return View();
}
// GET: EventCrudController/Create
0 references
public ActionResult Create()
{
return View();
}
// POST: EventCrudController/Create
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public ActionResult Create(IFormCollection collection)
{
try
{
return RedirectToAction(nameof(Index));
90 %
No issues found
Ln: 1
Ch: 1
SPC
CRLF
Output
Ready
Type here to search
28°C Light rain
ENG
11:28
30-09-2021


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Crud_Event
Debug
Any CPU
IIS Express
Live Share
EXP
EventCrudController.cs
EventMasterClass.cs*
ConnectionString.cs*
Crud_Event.csproj*
Crud_Event
Crud_Event.Models.EventMasterClass
E_Type_Event
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
namespace Crud_Event.Models
{
0 references
public class EventMasterClass
{
0 references
public int E_Id { get; set; }
0 references
public string E_Type_Event { get; set; }
0 references
public string E_Estimated_Cost { get; set; }
0 references
public string E_Actual_Cost { get; set; }
0 references
public string E_Date { get; set; }
}
}
90 %
No issues found
Ln: 15
Ch: 1
SPC
CRLF
Output
Ready
Type here to search
28°C Light rain
ENG
11:28
30-09-2021



EventCrudController.cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Crud_Event.Models;


namespace Crud_Event.Controllers
{


    public class EventCrudController : Controller
    {

        EventDataAccessLayer eventDataAccessLayer = new EventDataAccessLayer();
        // GET: EventCrudController
        public ActionResult Index()
        {
            EventDataAccessLayer eventDataAccessLayer = new EventDataAccessLayer();

            IEnumerable<EventMasterClass> ev = eventDataAccessLayer.GetAllEvent();
            return View(ev);
        }

        // GET: EventCrudController/Details/5
        public ActionResult Details(int id)
        {
            return View();
        }

        // GET: EventCrudController/Create
```

```
        public ActionResult Create()
        {
            return View();
        }

        // POST: EventCrudController/Create
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create(IFormCollection collection)
        {
            try
            {
                return RedirectToAction(nameof(Index));
            }
            catch
            {
                return View();
            }
        }

        // GET: EventCrudController/Edit/5
        public ActionResult Edit(int id)
        {
            return View();
        }

        // POST: EventCrudController/Edit/5
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Edit(int id, IFormCollection collection)
        {
            try
            {
                return RedirectToAction(nameof(Index));
            }
            catch
            {
                return View();
            }
        }

        // GET: EventCrudController/Delete/5
        public ActionResult Delete(int id)
        {
            return View();
        }

        // POST: EventCrudController/Delete/5
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Delete(int id, IFormCollection collection)
        {
            try
            {
                return RedirectToAction(nameof(Index));
            }
            catch
            {
                return View();
            }
        }
    }
```

```
}
```

EventDataAccessLayer.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Threading.Tasks;
using Event_CRUD.Utility;


namespace Crud_Event.Models
{
    public class EventDataAccessLayer
    {
       string  connectionString = ConnectionString.CName;
        public IEnumerable<EventMasterClass> GetAllEvent()
        {
            List<EventMasterClass> lstStudent = new List<EventMasterClass>();
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand("", con);
                cmd.CommandType = CommandType.StoredProcedure;
                con.Open();
                SqlDataReader rdr = cmd.ExecuteReader();

                while (rdr.Read())
                {
                    EventMasterClass ev = new EventMasterClass();
                    ev.E_Id = Convert.ToInt32(rdr["E_Id"]);
                    ev.E_Title = rdr["E_Title"].ToString();
                    ev.E_Type_Event = rdr["E_Type_Event"].ToString();
                    ev.E_Estimated_Cost = rdr["E_Estimated_Cost"].ToString();
                    ev.E_Actual_Cost = rdr["E_Actual_Cost"].ToString();
                    ev.E_Date = rdr["E_Date"].ToString();


                    lstStudent.Add(ev);
                }
                con.Close();
            }
            return lstStudent;
        }
        public void AddEvent(EventMasterClass ev)
        {
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand("", con);
                cmd.CommandType = CommandType.StoredProcedure;

                cmd.Parameters.AddWithValue("@E_Id", ev.E_Title);
                cmd.Parameters.AddWithValue("@E_Type_Event", ev.E_Type_Event);
                cmd.Parameters.AddWithValue("@E_Estimated_Cost", ev.E_Estimated_Cost);
                cmd.Parameters.AddWithValue("@E_Actual_Cost", ev.E_Actual_Cost);
                cmd.Parameters.AddWithValue("@E_Date", ev.E_Date);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
            }
        }
        public void UpdateEvent(EventMasterClass ev)
```

```
        {
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand("", con);
                cmd.CommandType = CommandType.StoredProcedure;

                cmd.Parameters.AddWithValue("@E_Id", ev.E_Title);
                cmd.Parameters.AddWithValue("@E_Type_Event", ev.E_Type_Event);
                cmd.Parameters.AddWithValue("@E_Estimated_Cost", ev.E_Estimated_Cost);
                cmd.Parameters.AddWithValue("@E_Actual_Cost", ev.E_Actual_Cost);
                cmd.Parameters.AddWithValue("@E_Date", ev.E_Date);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
            }
        }
        public EventMasterClass GetEvent(int? id)
        {
            EventMasterClass ev = new EventMasterClass();

            using (SqlConnection con = new SqlConnection(connectionString))
            {
                string sqlQuery = "SELECT * FROM Event_Master WHERE E_Id= " + id;
                SqlCommand cmd = new SqlCommand(sqlQuery, con);
                con.Open();
                SqlDataReader rdr = cmd.ExecuteReader();

                while (rdr.Read())
                {

                    ev.E_Id = Convert.ToInt32(rdr["E_Id"]);
                    ev.E_Title = rdr["E_Title"].ToString();
                    ev.E_Type_Event = rdr["E_Type_Event"].ToString();
                    ev.E_Estimated_Cost = rdr["E_Estimated_Cost"].ToString();
                    ev.E_Actual_Cost = rdr["E_Actual_Cost"].ToString();
                    ev.E_Date = rdr["E_Date"].ToString();
                }
            }
            return ev;
        }
        public void DeleteEvent(int? id)
        {
            using (SqlConnection con = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand("", con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@E_Id", id);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
            }
        }
    }
}
```

EventMasterClass.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
```

```
namespace Crud_Event.Models
{
    public class EventMasterClass
    {
        public int E_Id { get; set; }
        public string E_Title { get; set; }
        public string E_Type_Event { get; set; }
        public string E_Estimated_Cost { get; set; }
        public string E_Actual_Cost { get; set; }
        public string E_Date { get; set; }

    }
}
```

ConnectionString.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Event_CRUD.Utility
{
    public class ConnectionString
    {
        private static string cName = "Data Source=.; Initial Catalog=Events_DB;User
ID=sa;Password=123";
        public static string CName
        {
            get => cName;
        }
    }
}
```

Index.chtml

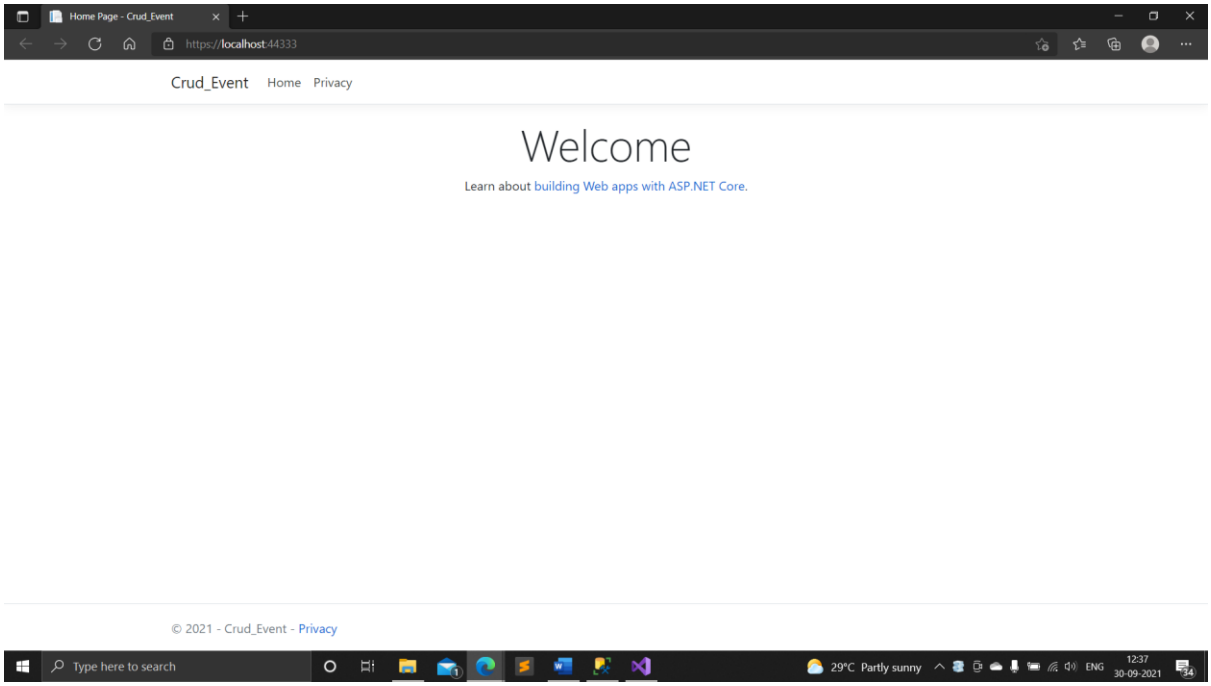
```
@model IEnumerable<Crud_Event.Models.EventMasterClass>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.E_Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.E_Title)
```

```
            </th>
            <th>
                @Html.DisplayNameFor(model => model.E_Type_Event)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.E_Estimated_Cost)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.E_Actual_Cost)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.E_Date)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.E_Id)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.E_Title)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.E_Type_Event)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.E_Estimated_Cost)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.E_Actual_Cost)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.E_Date)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ }) |
                @Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey */
}) |
                @Html.ActionLink("Delete", "Delete", new { /* id=item.PrimaryKey */ })
            </td>
        </tr>
}
    </tbody>
</table>
```


Home Page - Crud_Event
https://localhost:44333
Crud_Event
Home
Privacy
Welcome
Learn about building Web apps with ASP.NET Core.
© 2021 - Crud_Event - Privacy
Type here to search
29°C Partly sunny
ENG
12:37
30-09-2021