

Question 1

Output

Event List - Practical3_1

localhost:55790/Event?search=&submit=Search

UTU Webmail : Inbox Facebook Instagram Pinterest - India Text to HTML LinkedIn: Log In or... Analytics Sitemaps - Bing Webmaster Tools Products Images ... Status Forever - Firefox Reading list

Practical3_1 Home Privacy

Index

Create New

Microsoft.AspNetCore.Mvc.Rendering.MvcForm) [Estimated Cost]])

| EventCode | Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent | |
|-----------|------------------|-------------|---------------|------------|-------------|---|
| 1 | Sport Event | Sport | 1000 | 1200 | 01-09-2021 | Edit Details Delete |
| 2 | Wedding Event | Wedding | 1500 | 1560 | 10-05-2021 | Edit Details Delete |
| 3 | Tech House Event | Tech | 1800 | 1890.5 | 25-06-2021 | Edit Details Delete |
| 4 | Cricket Event | Sport | 2500 | 2562.41 | 08-10-2021 | Edit Details Delete |
| 5 | Annual Event | Wedding | 2900 | 3000.52 | 05-12-2021 | Edit Details Delete |

The screenshot shows a Microsoft Windows desktop environment. A browser window titled "Event List - Practical3_1" is open, displaying a list of events. The browser's address bar shows the URL "localhost:55790/Event?search=1500&submit=Search". The page content includes a header "Index" and a "Create New" link. Below is a table with the following data:

| EventCode | Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent |
|-----------|---------------|-------------|---------------|------------|-------------|
| 2 | Wedding Event | Wedding | 1500 | 1560 | 10-05-2021 |

The browser's toolbar contains various icons for search, refresh, and links to social media and other websites. The status bar at the bottom right indicates the date as "30-Nov-2021" and the time as "11:37 AM".

© 2021 - Practical3_1 - Privacy

MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

Event List - Practical3_1

localhost:55790/Event?search=2900&submit=Search

UTU Webmail : Inbox Facebook Instagram Pinterest - India Text to HTML LinkedIn: Log In or... Analytics Sitemaps - Bing We... Products Images Status Forever - Fir... Reading list

Practical3_1 Home Privacy

Index

Create New

Microsoft.AspNetCore.Mvc.Rendering.MvcForm) { Estimated Cost 2900 } Search }}

| EventCode | Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent | |
|-----------|--------------|-------------|---------------|------------|-------------|---|
| 5 | Annual Event | Wedding | 2900 | 3000.52 | 05-12-2021 | Edit Details Delete |

© 2021 - Practical3_1 - Privacy

Create - Practical3_1

localhost:55790/Event/Create

UTU Webmail : Inbox Facebook Instagram Pinterest - India Text to HTML LinkedIn: Log In or... Analytics Sitemaps - Bing We... Products Images Status Forever - Fir... Reading list

Practical3_1 Home Privacy

Create

EventModel

EventCode

Title

TypeOfEvent

EstimatedCost

ActualCost

DateOfEvent

[Create](#)

[Back to List](#)

Type here to search

11:38 AM ENG 30-09-2021

MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows a web browser window titled "Details - Practical3_1". The address bar indicates the URL is "localhost:55790/Event/Details/2". The page content displays the following details for an event:

| EventCode | 2 |
|---------------|---------------|
| Title | Wedding Event |
| TypeOfEvent | Wedding |
| EstimatedCost | 1500 |
| ActualCost | 1560 |
| DateOfEvent | 10-05-2021 |

Below the table, there are links for "Edit" and "Back to List".

The screenshot shows a web browser window titled "Delete - Practical3_1". The address bar indicates the URL is "localhost:55790/Event/Delete/3". The page content asks the user if they are sure they want to delete the event, followed by the event details:

| EventCode | 3 |
|---------------|------------------|
| Title | Tech House Event |
| TypeOfEvent | Tech |
| EstimatedCost | 1800 |
| ActualCost | 1890.5 |
| DateOfEvent | 25-06-2021 |

Below the table, there is a red "Delete" button and a link for "Back to List".

The screenshot shows a web browser window with a blank white page. The address bar indicates the URL is "localhost:55790/Event/Delete/3". This represents the state of the application after the event has been successfully deleted.

MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Edit - Practical3_1
- Address Bar:** localhost:55790/Event/Edit/4
- Toolbar:** Back, Forward, Stop, Refresh, Home, Favorites, Help, etc.
- Page Content:**
 - Section Header:** Edit
 - Section Label:** EventModel
 - Form Fields:**
 - EventCode: 4
 - Title: Cricket Event
 - TypeOfEvent: Sport
 - EstimatedCost: 2500
 - ActualCost: 2562.41
 - DateOfEvent: 08-10-2021
 - Buttons:** Save, Back to List
- Taskbar:** Shows the Windows Start button, a search bar with "Type here to search", and various pinned application icons (File Explorer, Edge, Mail, etc.).
- System Tray:** Displays the date (30-09-2021), time (11:38 AM), battery level, and network status.

Work Screenshots

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Code Editor:** The main window displays the `Index.cshtml` file, which contains C# Razor syntax for rendering a list of events. It includes sections for the title, a search bar, and a table to list the events.
- Solution Explorer:** On the right, the Solution Explorer shows the project structure for "Practical3_1". It includes files like `EventController.cs`, `EventModel.cs`, `HomeController.cs`, `ErrorViewModel.cs`, `EventModel.cs` (under Views), and various view files like `_Layout.cshtml` and `Index.cshtml`.
- Task List:** Below the Solution Explorer, the Task List shows "No issues found".
- Toolbars and Status Bar:** The top and bottom bars provide standard Windows-style controls and status information, including the date and time (30-09-2021).

EventModel.cs

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

namespace Practical3_1.Models
{
    public class EventModel
    {
        public int EventCode { get; set; }

        public string Title { get; set; }

        public string TypeOfEvent { get; set; }

        public float EstimatedCost { get; set; }

        public float ActualCost { get; set; }

        public string DateOfEvent { get; set; }
    }
}
```

EventController.cs

```
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Practical3_1.Models;

namespace Practical3_1.Controllers
{
    public class EventController : Controller
    {
        private static List<EventModel> events = new List<EventModel>();

        public EventController()
        {

            if (events.Count == 0)
            {
                events.Add(new EventModel { EventCode = 1, Title = "Sport Event",
TypeOfEvent = "Sport", EstimatedCost = 1000f, ActualCost = 1200f, DateOfEvent = "01-09-2021" });
                events.Add(new EventModel { EventCode = 2, Title = "Wedding Event",
TypeOfEvent = "Wedding", EstimatedCost = 1500f, ActualCost = 1560f, DateOfEvent = "10-05-2021" });
            }
        }
    }
}
```

```
        events.Add(new EventModel { EventCode = 3, Title = "Tech House Event",
TypeOfEvent = "Tech", EstimatedCost = 1800f, ActualCost = 1890.50f, DateOfEvent = "25-
06-2021" });
        events.Add(new EventModel { EventCode = 4, Title = "Creccket Event",
TypeOfEvent = "Sport", EstimatedCost = 2500f, ActualCost = 2562.41f, DateOfEvent =
"08-10-2021" });
        events.Add(new EventModel { EventCode = 5, Title = "Annual Event",
TypeOfEvent = "Wedding", EstimatedCost = 2900f, ActualCost = 3000.52f, DateOfEvent =
"05-12-2021" });
    }

}

// GET: EventController
public ActionResult Index(string search)
{
    if (search != null)
    {
        return View(events.Where(e => e.EstimatedCost ==
Convert.ToDouble(search)).ToList());
    }

    return View(events.ToList());
}

// GET: EventController/Details/5
public ActionResult Details(int id)
{
    return View(events.SingleOrDefault(e => e.EventCode == id));
}

// GET: EventController/Create
public ActionResult Create()
{
    return View();
}

// POST: EventController/Create
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([FromBody] EventModel newEvent)
{
    try
    {
        events.Add(newEvent);

        return RedirectToAction(nameof(Index));
    }
    catch
    {
        return View();
    }
}

// GET: EventController/Edit/5
public ActionResult Edit(int id)
{
    return View(events.SingleOrDefault(e => e.EventCode == id));
}

// POST: EventController/Edit/5
[HttpPost]
```

```
[ValidateAntiForgeryToken]
public ActionResult Edit(int id, [FromBody] EventModel updatedEvent)
{
    try
    {
        EventModel eobj = events.SingleOrDefault(e => e.EventCode == id);
        if (eobj != null && updatedEvent != null)
        {
            eobj.Title = updatedEvent.Title;
            eobj.TypeOfEvent = updatedEvent.TypeOfEvent;
            eobj.EstimatedCost = updatedEvent.EstimatedCost;
            eobj.ActualCost = updatedEvent.ActualCost;
            eobj.DateOfEvent = updatedEvent.DateOfEvent;
        }

        return RedirectToAction(nameof(Index));
    }
    catch
    {
        return View();
    }
}

// GET: EventController/Delete/5
public ActionResult Delete(int id)
{
    return View(events.SingleOrDefault(e => e.EventCode == id));
}

// POST: EventController/Delete/5
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Delete(int id, IFormCollection collection)
{
    try
    {
        EventModel eobj = events.SingleOrDefault(e => e.EventCode == id);
        if (eobj != null)
        {
            events.Remove(eobj);
        }

        return RedirectToAction(nameof(Index));
    }
    catch
    {
        return View();
    }
}
}
```

Create.cshtml

```
@model Practical3_1.Models.EventModel
@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
```

```
}
```

```
<h1>Create</h1>
```

```
<h4>EventModel</h4>
```

```
<hr />
```

```
<div class="row">
```

```
    <div class="col-md-4">
```

```
        <form asp-action="Create">
```

```
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
```

```
            <div class="form-group">
```

```
                <label asp-for="EventCode" class="control-label"></label>
```

```
                <input asp-for="EventCode" class="form-control" />
```

```
                <span asp-validation-for="EventCode" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="Title" class="control-label"></label>
```

```
                <input asp-for="Title" class="form-control" />
```

```
                <span asp-validation-for="Title" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="TypeOfEvent" class="control-label"></label>
```

```
                <input asp-for="TypeOfEvent" class="form-control" />
```

```
                <span asp-validation-for="TypeOfEvent" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="EstimatedCost" class="control-label"></label>
```

```
                <input asp-for="EstimatedCost" class="form-control" />
```

```
                <span asp-validation-for="EstimatedCost" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="ActualCost" class="control-label"></label>
```

```
                <input asp-for="ActualCost" class="form-control" />
```

```
                <span asp-validation-for="ActualCost" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <label asp-for="DateOfEvent" class="control-label"></label>
```

```
                <input asp-for="DateOfEvent" class="form-control" />
```

```
                <span asp-validation-for="DateOfEvent" class="text-danger"></span>
```

```
            </div>
```

```
            <div class="form-group">
```

```
                <input type="submit" value="Create" class="btn btn-primary" />
```

```
            </div>
```

```
        </form>
```

```
    </div>
```

```
</div>
```

```
    <a asp-action="Index">Back to List</a>
```

```
</div>
```

```
@section Scripts {
```

```
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
```

```
}
```

Delete.cshtml

```
@model Practical3_1.Models.EventModel
```

```
@{
    ViewData["Title"] = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>EventModel</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.EventCode)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.EventCode)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Title)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Title)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.TypeOfEvent)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.TypeOfEvent)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.EstimatedCost)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.EstimatedCost)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.ActualCost)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.ActualCost)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.DateOfEvent)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.DateOfEvent)
        </dd>
    </dl>

    <form asp-action="Delete">
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
```

Details.cshtml

```
@model Practical3_1.Models.EventModel

{@
    ViewData["Title"] = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h1>Details</h1>

<div>
    <h4>EventModel</h4>
    <hr />
    <dl class="row">
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.EventCode)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.EventCode)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.Title)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.Title)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.TypeOfEvent)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.TypeOfEvent)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.EstimatedCost)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.EstimatedCost)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.ActualCost)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.ActualCost)
        </dd>
        <dt class = "col-sm-2">
            @Html.DisplayNameFor(model => model.DateOfEvent)
        </dt>
        <dd class = "col-sm-10">
            @Html.DisplayFor(model => model.DateOfEvent)
        </dd>
    </dl>
</div>
<div>
    @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
    <a asp-action="Index">Back to List</a>
</div>
```

Edit.cshtml

```
@model Practical3_1.Models.EventModel

@{
    ViewData["Title"] = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h1>Edit</h1>

<h4>EventModel</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="EventCode" class="control-label"></label>
                <input asp-for="EventCode" class="form-control" />
                <span asp-validation-for="EventCode" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Title" class="control-label"></label>
                <input asp-for="Title" class="form-control" />
                <span asp-validation-for="Title" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="TypeOfEvent" class="control-label"></label>
                <input asp-for="TypeOfEvent" class="form-control" />
                <span asp-validation-for="TypeOfEvent" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="EstimatedCost" class="control-label"></label>
                <input asp-for="EstimatedCost" class="form-control" />
                <span asp-validation-for="EstimatedCost" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="ActualCost" class="control-label"></label>
                <input asp-for="ActualCost" class="form-control" />
                <span asp-validation-for="ActualCost" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="DateOfEvent" class="control-label"></label>
                <input asp-for="DateOfEvent" class="form-control" />
                <span asp-validation-for="DateOfEvent" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

}

Index.cshtml

```

@model IEnumerable<Practical3_1.Models.EventModel>

@{
    ViewData["Title"] = "Event List";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>

@Html.BeginForm("Index", "Event", FormMethod.Get)
{
    <text>Estimated Cost</text> @Html.TextBox("search")
    <input type="submit"
        name="submit"
        value="Search" />
}
}

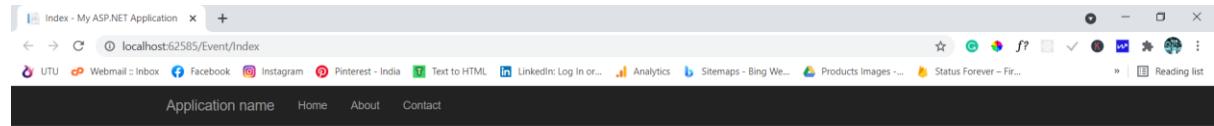
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.EventCode)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Title)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.TypeOfEvent)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.EstimatedCost)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.ActualCost)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.DateOfEvent)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.EventCode)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Title)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.TypeOfEvent)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.EstimatedCost)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.ActualCost)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.DateOfEvent)
                </td>
                <td></td>
            </tr>
        }
    </tbody>
</table>

```

```
</td>
<td>
    @Html.DisplayFor(modelItem => item.TypeOfEvent)
</td>
<td>
    @Html.DisplayFor(modelItem => item.EstimatedCost)
</td>
<td>
    @Html.DisplayFor(modelItem => item.ActualCost)
</td>
<td>
    @Html.DisplayFor(modelItem => item.DateOfEvent)
</td>
<td>
    @Html.ActionLink("Edit", "Edit", new { id = item.EventCode }) |
    @Html.ActionLink("Details", "Details", new { id = item.EventCode })
}) |
    @Html.ActionLink("Delete", "Delete", new { id = item.EventCode })
</td>
</tr>
}
</tbody>
</table>
```

Question 2

Output



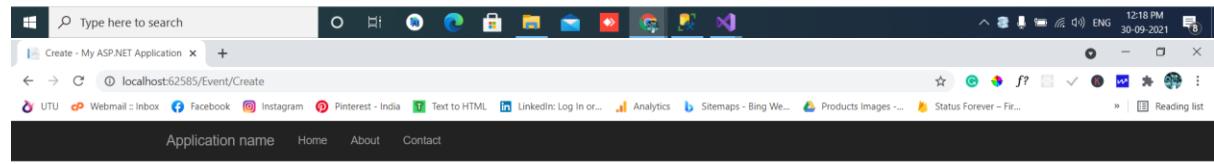
Index

Create New

System.Web.Mvc.Html.MvcForm { [GetEventAccordingType](#) | [EstimatedCostIsLessThanActualCost](#) | [EstimatedCostAndActualCostIsMoreThan50](#) }

| Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent | |
|-------------|-------------|---------------|------------|-------------|---|
| 31st Event | Annual | 500 | 90 | 02-01-2021 | Edit Details Delete |
| Sport Event | Sport | 200 | 250 | 08-05-2020 | Edit Details Delete |
| Game Event | Sport | 4000 | 4500.25 | 07-05-2021 | Edit Details Delete |
| Drawing | Competition | 5000 | 5500.45 | 31-12-2021 | Edit Details Delete |
| Photo Event | photography | 400 | 500 | 01-01-2020 | Edit Details Delete |
| ABC | Temp | 600 | 650 | 04-02-2021 | Edit Details Delete |

© 2021 - My ASP.NET Application



Create

tblEvent

| | |
|---------------|--|
| Title | <input type="text" value="Bicycle Race"/> |
| TypeOfEvent | <input type="text" value="Monthly Enjoy"/> |
| EstimatedCost | <input type="text" value="800"/> |
| ActualCost | <input type="text" value="1500"/> |
| DateOfEvent | <input type="text" value="30-09-2021"/> |

[Create](#)

[Back to List](#)

© 2021 - My ASP.NET Application



The record has been successfully created.

MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows a Microsoft Edge browser window with the title "Index - My ASP.NET Application". The address bar displays "localhost:62585/Event". The page content is titled "Index" and includes a "Create New" button. Below it is a table with columns: Title, TypeOfEvent, EstimatedCost, ActualCost, and DateOfEvent. The table contains seven rows of event data. At the bottom of the page is a copyright notice: "© 2021 - My ASP.NET Application".

| Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent | |
|--------------|---------------|---------------|------------|-------------|---|
| 31st Event | Annual | 500 | 90 | 02-01-2021 | Edit Details Delete |
| Sport Event | Sport | 200 | 250 | 08-05-2020 | Edit Details Delete |
| Game Event | Sport | 4000 | 4500.25 | 07-05-2021 | Edit Details Delete |
| Drawing | Competition | 5000 | 5500.45 | 31-12-2021 | Edit Details Delete |
| Photo Event | photography | 400 | 500 | 01-01-2020 | Edit Details Delete |
| ABC | Temp | 600 | 650 | 04-02-2021 | Edit Details Delete |
| Bicycle Race | Monthly Enjoy | 800 | 1500 | 30-09-2021 | Edit Details Delete |

The screenshot shows a Microsoft Edge browser window with the title "Details - My ASP.NET Application". The address bar displays "localhost:62585/Event/Details/5". The page content is titled "Details" and includes a heading "tblEvent". Below it is a table with columns: Title, TypeOfEvent, EstimatedCost, ActualCost, and DateOfEvent. The table contains one row of event data. At the bottom of the page is a copyright notice: "© 2021 - My ASP.NET Application".

| Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent |
|-------------|-------------|---------------|------------|-------------|
| Photo Event | photography | 400 | 500 | 01-01-2020 |



MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows a Microsoft Edge browser window with the title "Delete - My ASP.NET Application". The URL in the address bar is "localhost:62585/Event/Delete/4". The page content asks "Are you sure you want to delete this? tblEvent". Below this, there is a table with the following data:

| Title | Drawing |
|---------------|-------------|
| TypeOfEvent | Competition |
| EstimatedCost | 5000 |
| ActualCost | 5500.45 |
| DateOfEvent | 31-12-2021 |

At the bottom left is a "Delete" button, and at the bottom right is a "Back to List" link. The footer of the page says "© 2021 - My ASP.NET Application".

The screenshot shows a Microsoft Edge browser window with the title "Edit - My ASP.NET Application". The URL in the address bar is "localhost:62585/Event/Edit/3". The page content is titled "tblEvent". It contains a form with the following fields:

| | |
|---------------|------------|
| Title | Game Event |
| TypeOfEvent | Sport |
| EstimatedCost | 4000 |
| ActualCost | 4500.25 |
| DateOfEvent | 07-05-2021 |

Below the form is a "Save" button. At the bottom left is a "Back to List" link. The footer of the page says "© 2021 - My ASP.NET Application".



Advance search functionality based on following criteria

Index

Create New

| Title | TypeOfEvent | EstimatedCost | ActualCost | DateOfEvent | |
|--------------|---------------|---------------|------------|-------------|-------------------------|
| 31st Event | Annual | 500 | 90 | 02-01-2021 | Edit Details Delete |
| Drawing | Competition | 5000 | 5500.45 | 31-12-2021 | Edit Details Delete |
| Bicycle Race | Monthly Enjoy | 800 | 1500 | 30-08-2021 | Edit Details Delete |
| Photo Event | photography | 400 | 500 | 01-01-2020 | Edit Details Delete |
| Sport Event | Sport | 200 | 250 | 08-05-2020 | Edit Details Delete |
| Game Event | Sport | 4000 | 4500.25 | 07-05-2021 | Edit Details Delete |
| ABC | Temp | 600 | 650 | 04-02-2021 | Edit Details Delete |

© 2021 - My ASP.NET Application



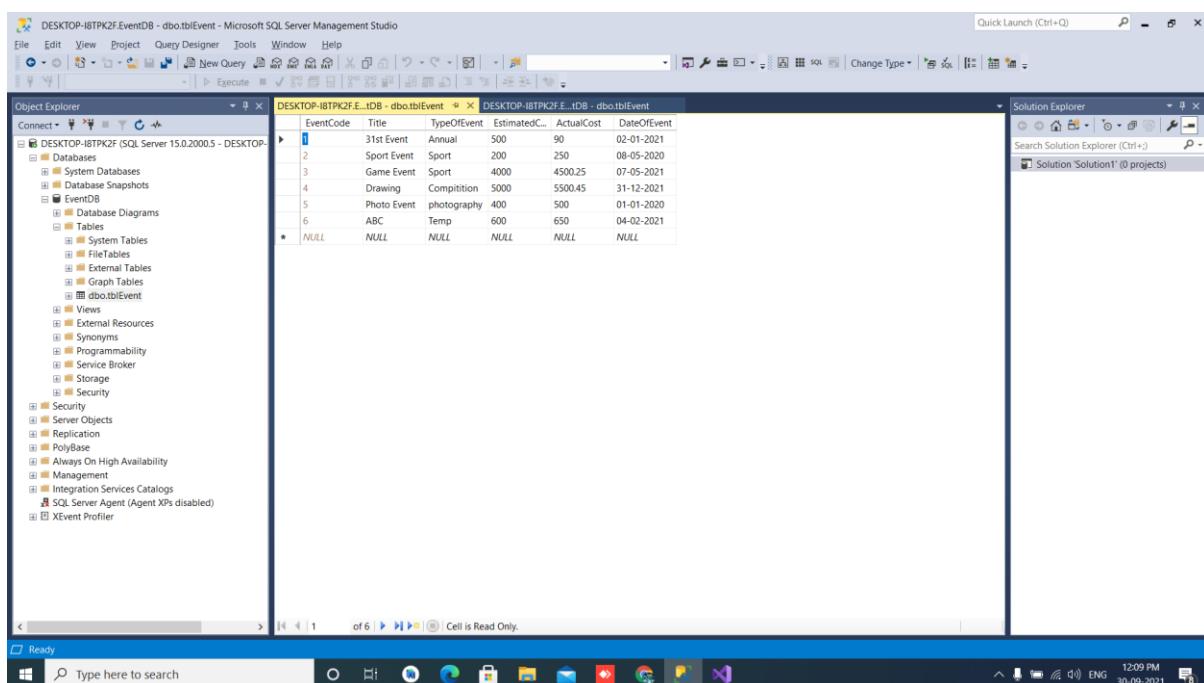
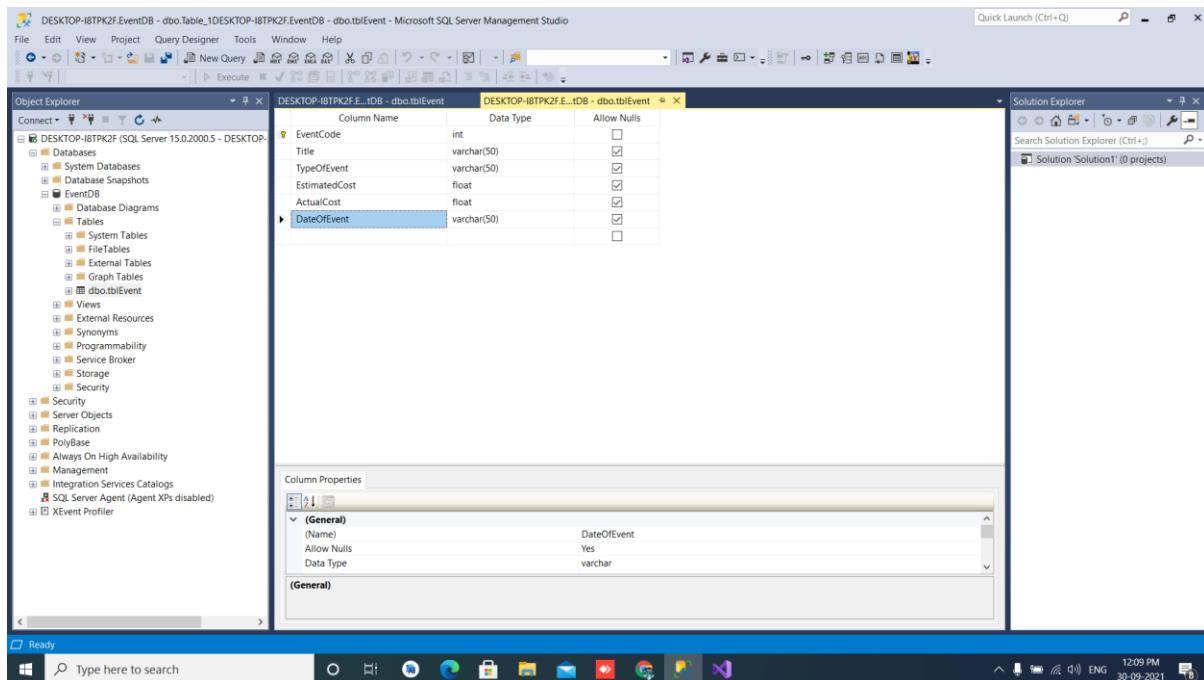
MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows a Microsoft Edge browser window with the title "Index - My ASP.NET Application". The address bar displays "localhost:62585/Event?submit=EstimatedCostAndActualCostsMoreThan500". The page content includes a navigation bar with links for "Application name", "Home", "About", and "Contact". Below the navigation bar is a section titled "Index" with a "Create New" link. A table lists event data with columns: Title, TypeOfEvent, EstimatedCost, ActualCost, and DateOfEvent. The table contains three rows: "Game Event" (Sport, 4000, 4500.25, 07-05-2021), "Drawing" (Competition, 5000, 5500.45, 31-12-2021), and "Bicycle Race" (Monthly Enjoy, 800, 1500, 30-09-2021). Each row has "Edit | Details | Delete" links. At the bottom of the page is a copyright notice: "© 2021 - My ASP.NET Application".

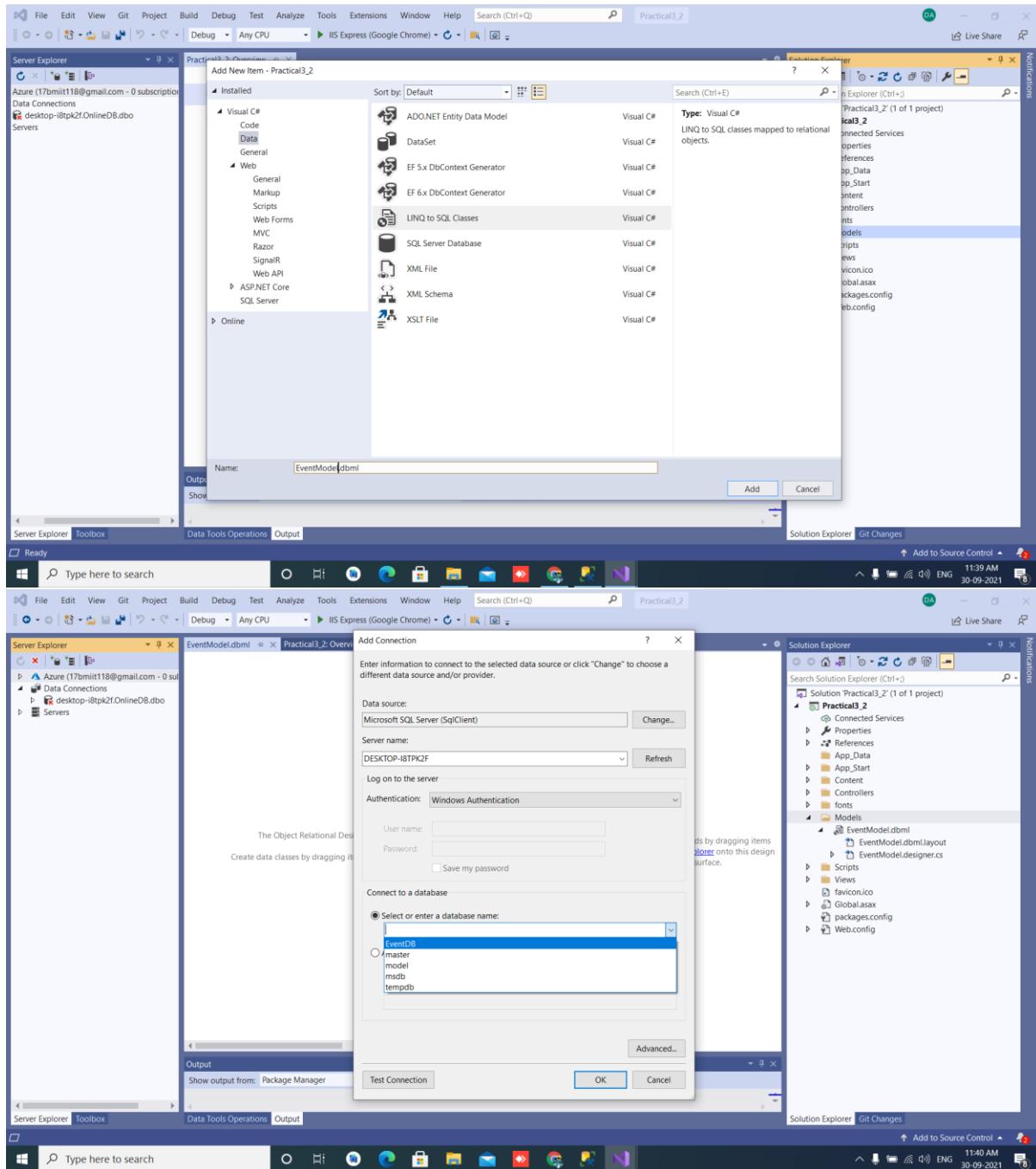
The screenshot shows a Microsoft Edge browser window with the title "Index - My ASP.NET Application". The address bar displays "localhost:62585/Event?submit=EstimatedCostsLessThanActualCost". The page content includes a navigation bar with links for "Application name", "Home", "About", and "Contact". Below the navigation bar is a section titled "Index" with a "Create New" link. A table lists event data with columns: Title, TypeOfEvent, EstimatedCost, ActualCost, and DateOfEvent. The table contains seven rows: "Sport Event" (Sport, 200, 250, 08-05-2020), "Game Event" (Sport, 4000, 4500.25, 07-05-2021), "Drawing" (Competition, 5000, 5500.45, 31-12-2021), "Photo Event" (photography, 400, 500, 01-01-2020), "ABC" (Temp, 600, 650, 04-02-2021), and "Bicycle Race" (Monthly Enjoy, 800, 1500, 30-09-2021). Each row has "Edit | Details | Delete" links. At the bottom of the page is a copyright notice: "© 2021 - My ASP.NET Application".



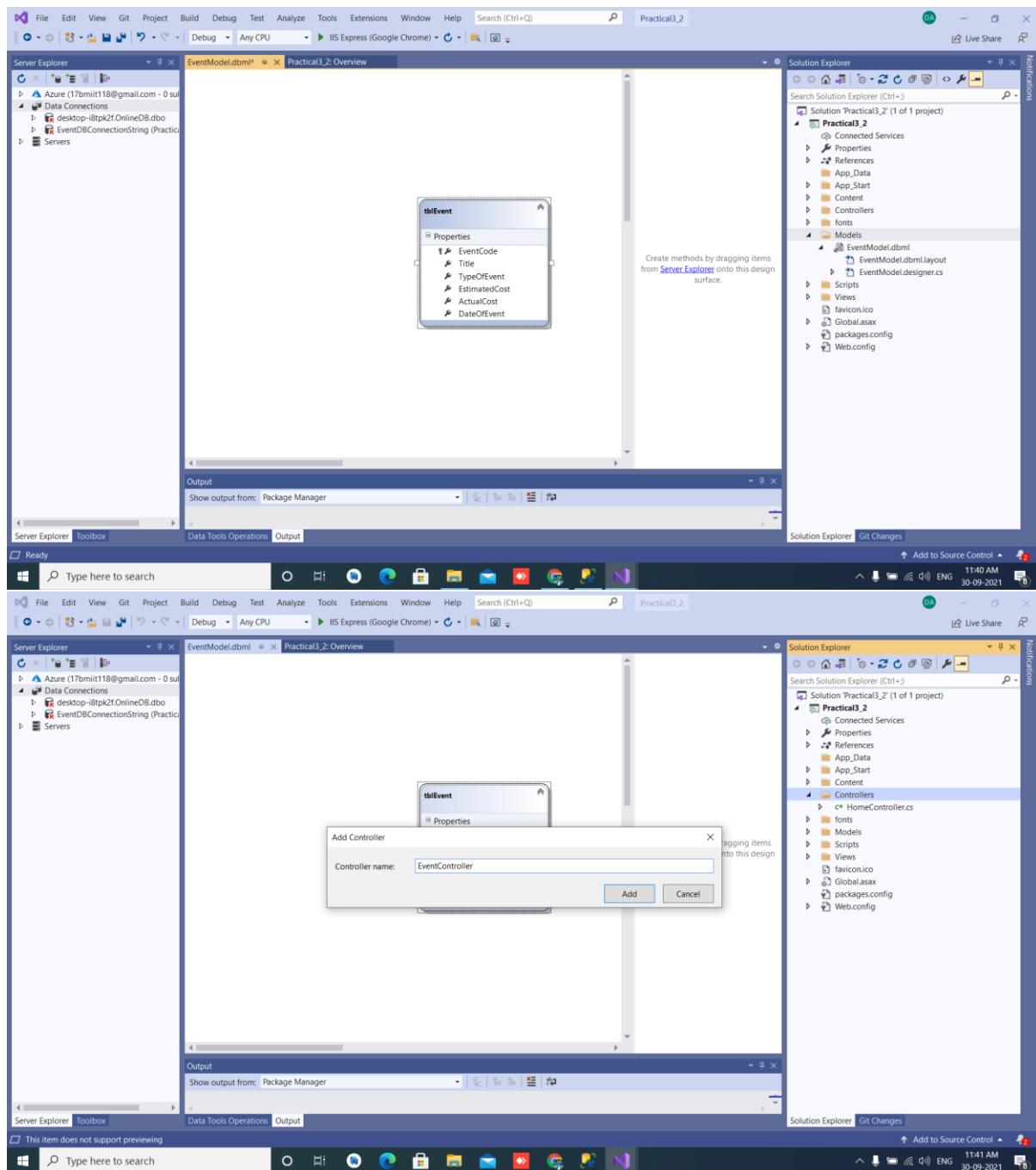
Database Screenshot



Work Screenshot



MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)



MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows two instances of Microsoft Visual Studio side-by-side.

Top Window: The title bar says "EventController.cs" and "Practical3_2". The code editor displays the following C# code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Practical3_2.Models;

namespace Practical3_2.Controllers
{
    public class EventController : Controller
    {
        // GET: Event
        public ActionResult Index()
        {
            return View();
        }

        // GET: Event/Create
        public ActionResult Create()
        {
            return View();
        }
    }
}
```

A modal dialog titled "Add View" is open, showing the following options:

- View name: Index
- Template: Empty (without model)
- Model class: Event
- Options:
 - Create
 - Delete
 - Details
 - Edit
 - Empty
 - Reference
 - Use a layout
 - List
- (Leave empty if it is set in a Razor _viewstart file)

Bottom Window: The title bar says "EventController.cs" and "Practical3_2". The code editor displays the following C# code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Practical3_2.Models;

namespace Practical3_2.Controllers
{
    public class EventController : Controller
    {
        // POST: Event/Delete/
        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add delete logic here
                tblEvent newEvent = dbobj.tblEvents.where(x => x.EventCode == id).FirstOrDefault();
                dbobj.tblEvents.DeleteOnSubmit(newEvent);
                dbobj.SubmitChanges();
                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

The Solution Explorer on the right shows the project structure for "Practical3_2".

MSC.ICT (SEM - 2) (Practical Exam – 205) ROLL NO. 02 (DARSHIT ASALALIYA)

The screenshot shows the Microsoft Visual Studio interface. The code editor displays the `Index.cshtml` file from the `Practical3_2` project. The file contains C# Razor code for a controller action. The Solution Explorer on the right shows the project structure, including files like `EventController.cs`, `HomeController.cs`, `EventModel.dbml`, and various view files (`Create.cshtml`, `Delete.cshtml`, etc.). The status bar at the bottom indicates the date and time as 30-09-2021 12:05 PM.

```
12     entModelDataContext dbobj = new EventModelDataContext();
13
14     GET: Event
15     <!--: ActionResult Index(string submit)
16
17     if(submit != null)
18     {
19         if(submit == "GetEventAccordingType")
20         {
21             var orderByResult = from s in dbobj.tblEvents.ToList()
22                                 orderby s.TypeOfEvent
23                                 select s;
24
25             return View(orderByResult.ToList());
26         }
27         else if(submit == "EstimatedCostIsLessThanActualCost")
28         {
29             return View(dbobj.tblEvents.Where(x => x.EstimatedCost < x.ActualCost).ToList());
30         }
31         else if(submit == "EstimatedCostAndActualCostIsMoreThan500")
32         {
33             return View(dbobj.tblEvents.Where(x => (x.ActualCost - x.EstimatedCost) > 500).ToList());
34         }
35     }
36
37     return View(dbobj.tblEvents.ToList());
38
39
40     GET: Event/Details/5
41
42
43
44
45
46
47
48
```

EventModel.dbml

The screenshot shows the Microsoft Visual Studio interface with the EntityDataSource Designer open. The designer window displays the `tblEvent` entity with its properties: `EventCode`, `Title`, `TypeOfEvent`, `EstimatedCost`, `ActualCost`, and `DateOfEvent`. A tooltip suggests dragging items from the Server Explorer onto the design surface. The Solution Explorer on the right shows the project structure, including files like `EventModel.designer.cs`, `Index.cshtml`, `Delete.cshtml`, `Edit.cshtml`, `Create.cshtml`, and various view files. The status bar at the bottom indicates the date and time as 30-09-2021 12:21 PM.

EventController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using Practical3_2.Models;

namespace Practical3_2.Controllers
{
    public class EventController : Controller
    {
        EventModelDataContext dbobj = new EventModelDataContext();

        // GET: Event
        public ActionResult Index(string submit)
        {
            if (submit != null)
            {
                if (submit == "GetEventAccordingType")
                {
                    var orderByResult = from s in dbobj.tblEvents.ToList()
                                        orderby s.TypeOfEvent
                                        select s;

                    return View(orderByResult.ToList());
                }
                else if (submit == "EstimatedCostIsLessThanActualCost")
                {
                    return View(dbobj.tblEvents.Where(x => x.EstimatedCost < x.ActualCost).ToList());
                }
                else if (submit == "EstimatedCostAndActulaCostIsMoreThan500")
                {
                    return View(dbobj.tblEvents.Where(x => (x.ActualCost - x.EstimatedCost) > 500).ToList());
                }
            }

            return View(dbobj.tblEvents.ToList());
        }

        // GET: Event/Details/5
        public ActionResult Details(int id)
        {
            tblEvent eobj = dbobj.tblEvents.Where(x => x.EventCode == id).FirstOrDefault();
            return View(eobj);
        }

        // GET: Event/Create
        public ActionResult Create()
        {
            return View();
        }

        // POST: Event/Create
        [HttpPost]
        public ActionResult Create(tblEvent tempobj)
        {
            try
```

```
{  
    // TODO: Add insert logic here  
    dbobj.tblEvents.InsertOnSubmit(tempobj);  
    dbobj.SubmitChanges();  
    return RedirectToAction("Index");  
}  
catch  
{  
    return View();  
}  
}  
  
// GET: Event/Edit/5  
public ActionResult Edit(int id)  
{  
    return View(dbobj.tblEvents.Where(x => x.EventCode ==  
id).FirstOrDefault());  
}  
  
// POST: Event/Edit/5  
[HttpPost]  
public ActionResult Edit(int id, tblEvent tempobj)  
{  
    try  
    {  
        // TODO: Add update logic here  
        tblEvent newEvent = dbobj.tblEvents.Where(x => x.EventCode ==  
id).FirstOrDefault();  
        newEvent.Title = tempobj.Title;  
        newEvent.TypeOfEvent = tempobj.TypeOfEvent;  
        newEvent.EstimatedCost = tempobj.EstimatedCost;  
        newEvent.ActualCost = tempobj.ActualCost;  
        newEvent.DateOfEvent = tempobj.DateOfEvent;  
        dbobj.SubmitChanges();  
        return RedirectToAction("Index");  
    }  
    catch  
    {  
        return View();  
    }  
}  
  
// GET: Event/Delete/5  
public ActionResult Delete(int id)  
{  
    return View(dbobj.tblEvents.Where(x => x.EventCode ==  
id).FirstOrDefault());  
}  
  
// POST: Event/Delete/5  
[HttpPost]  
public ActionResult Delete(int id, FormCollection collection)  
{  
    try  
    {  
        // TODO: Add delete logic here  
        tblEvent newEvent = dbobj.tblEvents.Where(x => x.EventCode ==  
id).FirstOrDefault();  
        dbobj.tblEvents.DeleteOnSubmit(newEvent);  
        dbobj.SubmitChanges();  
        return RedirectToAction("Index");  
    }  
}
```

```
        catch
        {
            return View();
        }
    }
}
```

Index.cshtml

```
@model IEnumerable<Practical3_2.Models.tblEvent>

 @{
    ViewBag.Title = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>

@Html.BeginForm("Index", "Event", FormMethod.Get)
{
    <input type="submit"
           name="submit"
           value="GetEventAccordingType" />

    <input type="submit"
           name="submit"
           value="EstimatedCostIsLessThanActualCost" />

    <input type="submit"
           name="submit"
           value="EstimatedCostAndActualCostIsMoreThan500" />
}

<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Title)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.TypeOfEvent)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.EstimatedCost)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ActualCost)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.DateOfEvent)
        </th>
        <th></th>
    </tr>
```

```

@foreach (var item in Model)
{
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Title)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.TypeOfEvent)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.EstimatedCost)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ActualCost)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.DateOfEvent)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id = item.EventCode }) |
            @Html.ActionLink("Details", "Details", new { id = item.EventCode }) |
            @Html.ActionLink("Delete", "Delete", new { id = item.EventCode })
        </td>
    </tr>
}
</table>

```

Create.cshtml

```

@model Practical3_2.Models.tblEvent

 @{
    ViewBag.Title = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Create</h2>

@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>tblEvent</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Title, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Title, "", new { @class =
"text-danger" })
            </div>
        </div>
    </div>
}

```

```

        <div class="form-group">
            @Html.LabelFor(model => model.TypeOfEvent, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.TypeOfEvent, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.TypeOfEvent, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.EstimatedCost, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EstimatedCost, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EstimatedCost, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ActualCost, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.ActualCost, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.ActualCost, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.DateOfEvent, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.DateOfEvent, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.DateOfEvent, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Create" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Details.cshtml

```
@model Practical3_2.Models.tblEvent

@{
    ViewBag.Title = "Details";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Details</h2>

<div>
    <h4>tblEvent</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Title)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Title)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.TypeOfEvent)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.TypeOfEvent)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.EstimatedCost)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.EstimatedCost)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.ActualCost)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.ActualCost)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.DateOfEvent)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.DateOfEvent)
        </dd>

    </dl>
</div>
<p>
    @Html.ActionLink("Edit", "Edit", new { id = Model.EventCode }) |
    @Html.ActionLink("Back to List", "Index")

```

</p>

Edit.cshtml

```
@model Practical3_2.Models.tblEvent

@{
    ViewBag.Title = "Edit";
    Layout = "~/Views/Shared/_Layout.cshtml";
}



## Edit



@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>tblEvent</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.EventCode)

        <div class="form-group">
            @Html.LabelFor(model => model.Title, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Title, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Title, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.TypeOfEvent, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.TypeOfEvent, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.TypeOfEvent, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.EstimatedCost, htmlAttributes: new { @class
= "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.EstimatedCost, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.EstimatedCost, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">

```

```

        @Html.LabelFor(model => model.ActualCost, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.ActualCost, new { htmlAttributes = new
{ @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.ActualCost, "", new { @class
= "text-danger" })
        </div>
    </div>

    <div class="form-group">
        @Html.LabelFor(model => model.DateOfEvent, htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.DateOfEvent, new { htmlAttributes = new
{ @class = "form-control" } })
            @Html.ValidationMessageFor(model => model.DateOfEvent, "", new {
@class = "text-danger" })
        </div>
    </div>

    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit" value="Save" class="btn btn-default" />
        </div>
    </div>
</div>

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

Delete.cshtml

```

@model Practical3_2.Models.tblEvent

 @{
    ViewBag.Title = "Delete";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>tblEvent</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            @Html.DisplayNameFor(model => model.Title)
        </dt>
        <dd>
            @Html.DisplayFor(model => model.Title)
        </dd>
    </dl>
</div>

```

```
</dd>

<dt>
    @Html.DisplayNameFor(model => model.TypeOfEvent)
</dt>

<dd>
    @Html.DisplayFor(model => model.TypeOfEvent)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.EstimatedCost)
</dt>

<dd>
    @Html.DisplayFor(model => model.EstimatedCost)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.ActualCost)
</dt>

<dd>
    @Html.DisplayFor(model => model.ActualCost)
</dd>

<dt>
    @Html.DisplayNameFor(model => model.DateOfEvent)
</dt>

<dd>
    @Html.DisplayFor(model => model.DateOfEvent)
</dd>

</dl>

@using (Html.BeginForm()) {
    @Html.AntiForgeryToken()

    <div class="form-actions no-color">
        <input type="submit" value="Delete" class="btn btn-default" /> |
        @Html.ActionLink("Back to List", "Index")
    </div>
}
</div>
```