

Booking:

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Web;

namespace ASP_EF_App1.Models
{
    public class Booking
    {
        private int _bookingId;

        private string _customerId;

        private string _bookingDate;

        private double _amount = 0;

        [Required, DatabaseGenerated(DatabaseGeneratedOption.None)]
        [Key]
        public int BookingId
        {
            get { return _bookingId; }
            set { _bookingId = value; }
        }

        [Required]
        public string CustomerId
        {
            get { return _customerId; }
            set { _customerId = value; }
        }

        [Required]
        public string BookingDate
        {
            get { return _bookingDate; }
            set { _bookingDate = value; }
        }

        [Required]
        public double Amount
        {
            get { return _amount; }
            set { _amount = value; }
        }
    }
}
```

BookingEntity:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
```

```

using System.Linq;
using System.Web;

namespace ASP_EF_App1.Models
{
    public class BookingEntity : DbContext
    {
        public BookingEntity()
            : base("DataConnection")
        {
            Configuration.LazyLoadingEnabled = true;
            Database.SetInitializer<BookingEntity>(new
BookingEntityDBInitializer());
        }
        public DbSet<Booking> Bookings {get; set;}
    }

    public class BookingEntityDBInitializer :
DropCreateDatabaseAlways<BookingEntity>
    {
        protected override void Seed(BookingEntity context)
        {
            base.Seed(context);
        }
    }
}

```

HomeController:

```

using ASP_EF_App1.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;

namespace ASP_EF_App1.Controllers
{
    public class HomeController : Controller
    {
        private BookingEntity db = new BookingEntity();

        // GET: Home
        public ActionResult Display()
        {
            return View(db.Bookings.ToList());
        }

        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]

```

```

[ValidateAntiForgeryToken]
public ActionResult Index([Bind(Include =
"BookingId, CustomerId, BookingDate, Amount")] Booking booking)
{
    if (ModelState.IsValid)
    {
        db.Bookings.Add(booking);
        db.SaveChanges();
        return RedirectToAction("Display");
    }

    return View(booking);
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}
}

```

Display.cshtml:

```

@model IEnumerable<ASP_EF_App1.Models.Booking>
@{
    ViewBag.Title = "Display";
}
<h2>Display</h2>
<p>
    @Html.ActionLink("Create New", "Index")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.BookingId)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.CustomerId)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.BookingDate)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Amount)
        </th>
    </tr>
    @foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.BookingId)
        </td>
    <td>

```

```

        @Html.DisplayFor(modelItem => item.CustomerId)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.BookingDate)
    </td>
    <td>
        @Html.DisplayFor(modelItem => item.Amount)
    </td>
</tr>
}
</table>

```

Index.cshtml:

```
@model ASP_EF_App1.Models.Booking
```

```
@{
    ViewBag.Title = "Index";
}
```

```
<h2>Index</h2>
```

```

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

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

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

```

```

        @Html.ValidationMessageFor(model => model.BookingDate, "", new { @class
= "text-danger" })
    </div>
</div>

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

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

```