

ЗА ДЪРЖАВЕН ИЗПИТ  
– ЧАСТ ПО ПРАКТИКА НА ПРОФЕСИЯТА

## Списък със задачи

(Проектиране на уеб приложение с MVC архитектура)

Приложението трябва да държи **задачи**, които са основните обекти на приложението.

Задачите могат да променят статуса си:

- **Отворен**
- **В процес на изпълнение**
- **Завършен**

Функционалността на приложението трябва да поддържа **създаване, изброяване, редактиране и изтриване** на задачи.

Приложението трябва да **пази** данните в **база данни**.

## Модел на данните

Имплементирайте клас **Task** със следните три свойства:

- **id** – цяло число
- **title** – текст
- **status** – текст, който може да съдържа една от следните стойности:
  - **Open**
  - **In Progress**
  - **Finished**

## Скелет на проект

Скелетът има частично внедрена логика, така че ще трябва да напишете някакъв код, за да може приложението да заработи.

Изглед страниците ще ви бъдат предоставени изцяло. Трябва само да ги включите във вашата бизнес логика.

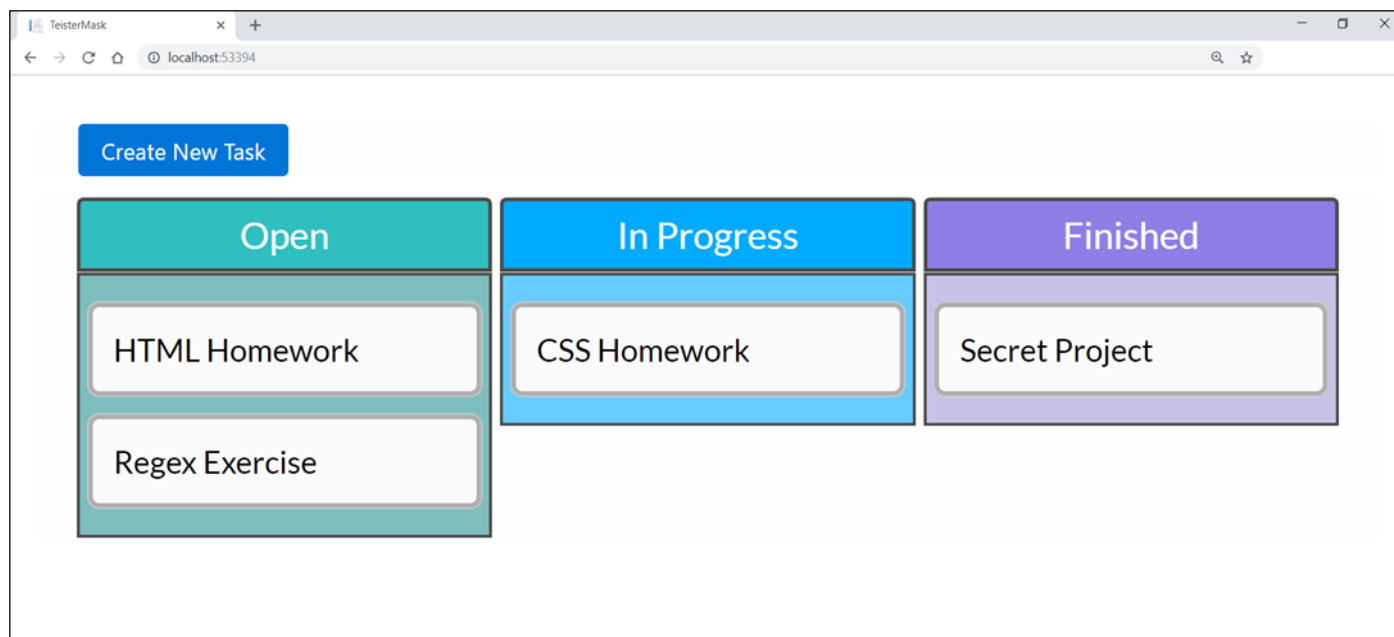
## Потребителски интерфейс

По-долу са описани как трябва да изглеждат страниците:

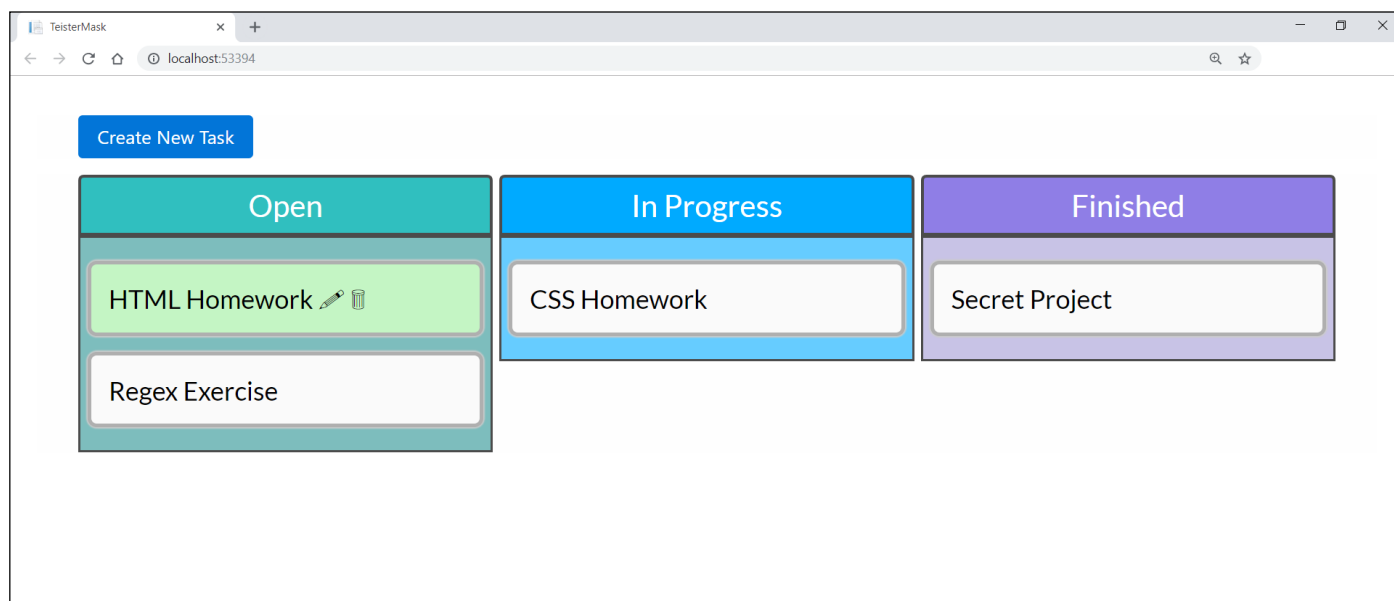
### Индексна страница

Всяка от трите колони съдържа само списък **само задачите**:

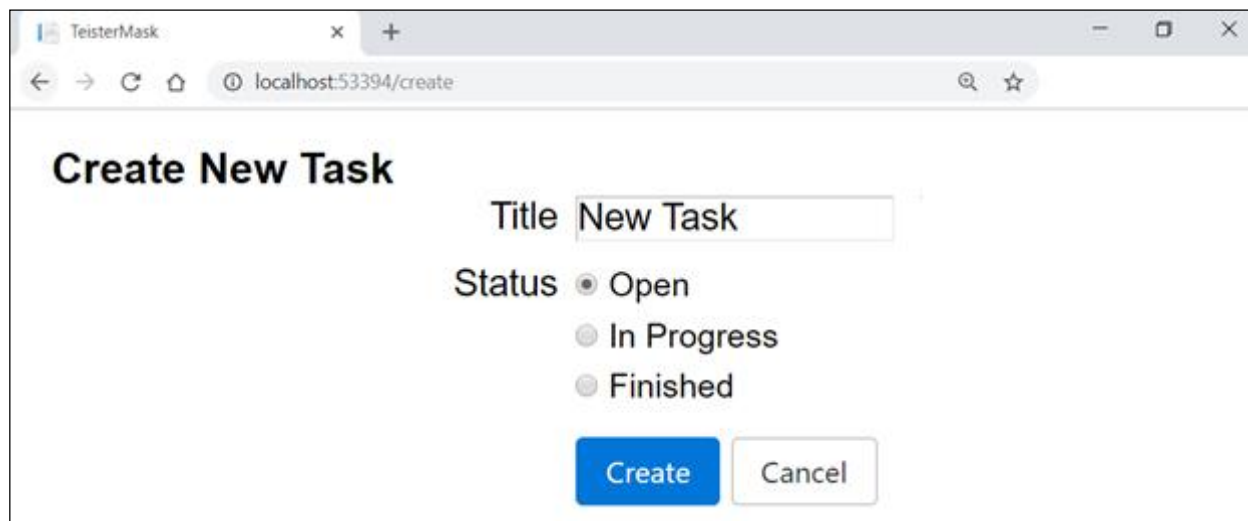
## ИНТЕРНЕТ ПРОГРАМИРАНЕ



Когато потребителят задържи курсора си над задача, бутонът за редактиране трябва да сочи към пътя `/Edit/{id}` :



## Страница за създаване на задача



TeisterMask x +

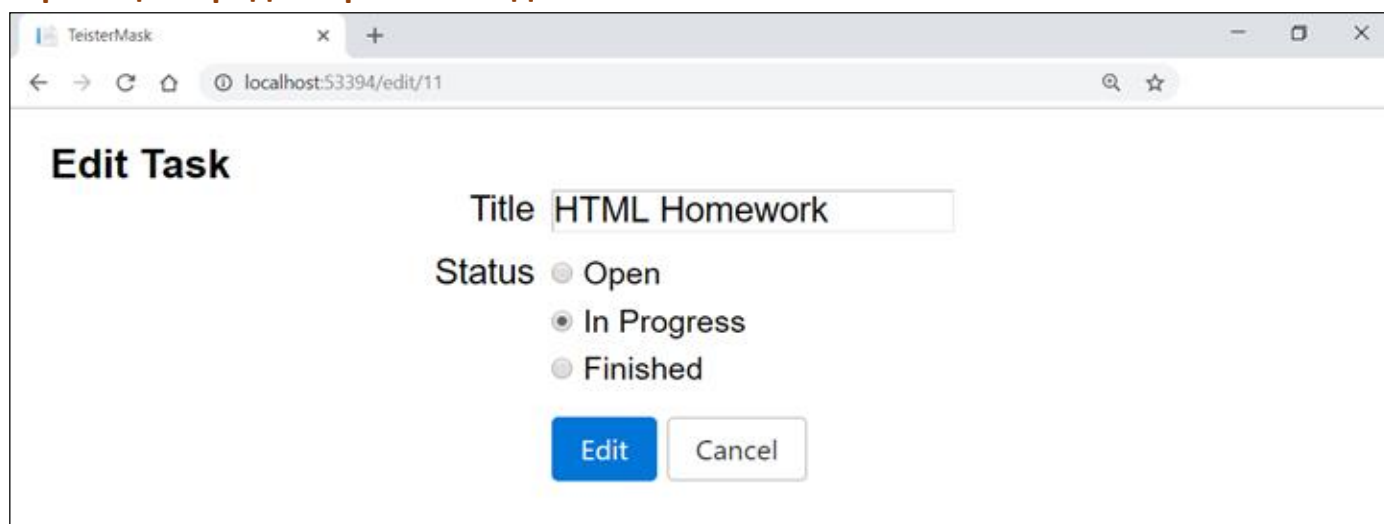
localhost:53394/create

### Create New Task

Title

Status ☒ Open  
☐ In Progress  
☐ Finished

## Страница за редактиране на задача



TeisterMask x +

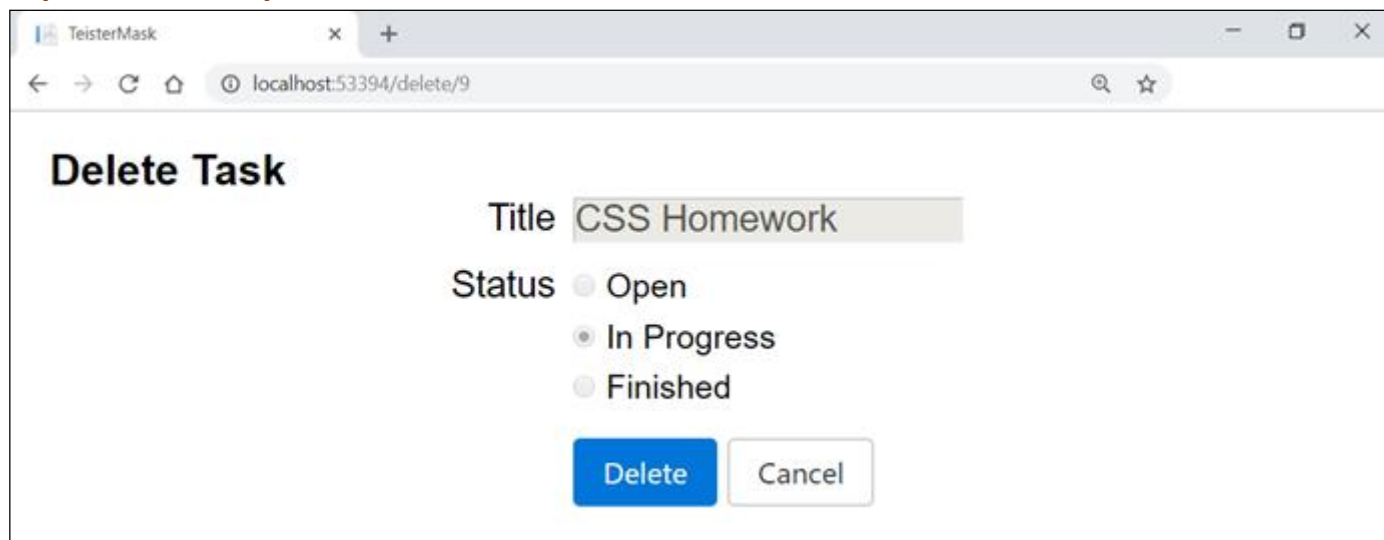
localhost:53394/edit/11

### Edit Task

Title

Status ☐ Open  
☒ In Progress  
☐ Finished

## Страница за изтриване на задача



TeisterMask x +

localhost:53394/delete/9

### Delete Task

Title

Status ☐ Open  
☒ In Progress  
☐ Finished

## ИНТЕРНЕТ ПРОГРАМИРАНЕ

1. Създайте база данни с една таблица с име Task

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Title	varchar(50)	<input type="checkbox"/>
Status	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

2. Създайте уеб приложение с архитектура MVC
3. Инсталирайте EntityFramework библиотеките и scaffold-нете базата данни в папка Models
4. Създайте контролер с име TaskController, за да управлявате изгледите е необходимо да напишете някакъв код, за да може приложението да заработи.
5. Изглед страниците ще ви бъдат предоставени изцяло. Трябва само да ги включите във вашата бизнес логика.

5.1 За файла \_Layout.cshtml

```
<html>
<head>
    <title>TeisterMask</title>
    @Styles.Render("~/Content/index-style.css")
    @Styles.Render("~/Content/form-style.css")
</head>
<body>
    @RenderBody()
</body>
</html>
```

5.2 За файла Index.cshtml

```
@model List<TeisterMask.Models.Task>

@{ ViewBag.Title = "Index"; }

<div class="container">
    @Html.ActionLink("Create New Task", "Create", "Task", null, new { @class = "button" })
</div>

<div class="tasks container">
    <header>
        <div class="column header open">Open</div>
        <div class="column header inProgress">In progress</div>
        <div class="column header finished">Finished</div>
    </header>
    <ul class="column open">
        @foreach (var task in Model.Where(t => t.Status == "Open").ToList())
        {
            <li>
                @task.Title
                @Html.ActionLink(" ", "Edit", "Task", new { id = task.Id }, new { @class = "icon edit" })
            </li>
        }
    </ul>
</div>
```

```

</li>}
</ul>
<ul class="column inProgress">
    @foreach (var task in Model.Where(t => t.Status == "In Progress").ToList())
    {
<li>
    @task.Title
    @Html.ActionLink(" ", "Edit", "Task", new { id = task.Id }, new { @class = "icon
edit" })
</li>}
</ul>
<ul class="column finished">
    @foreach (var task in Model.Where(t => t.Status == "Finished").ToList())
    {
<li>
    @task.Title
    @Html.ActionLink(" ", "Edit", "Task", new { id = task.Id }, new { @class = "icon
edit" })
</li>}
</ul>
</div>

```

### 5.3 За файла Create.cshtml

```

@model TeisterMask.Models.Task

@{ ViewBag.Title = "Create"; }

<header>
    <h3>Create New Task</h3>
</header>

@using (Html.BeginForm("Create", "Task", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div>
        <label class="desc" for="title">Title</label>
        <div>
            <input id="title" name="title" type="text" class="field text
fn" value="" size="8" tabindex="1" autofocus>
        </div>
    </div>

    <div>
        <fieldset>
            <legend id="title5" class="desc">
                Status
            </legend>

            <div>
                <div>
                    <input id="status1" name="status"
type="radio" value="Open" tabindex="2" checked>
                    <label class="choice"
for="status1">Open</label>
                </div>
                <div>
                    <input id="status2" name="status"
type="radio" value="In Progress" tabindex="3">
                    <label class="choice"
for="status2">In Progress</label>
                </div>
            </div>
        </div>
    </div>

```

```



```

## 5.4 За файла Edit.cshtml

```

@model TeisterMask.Models.Task
@{ ViewBag.Title = "Edit Task"; }

<header>
    <h3>Edit Task</h3>
</header>

@using (Html.BeginForm("Edit", "Task", FormMethod.Post))
{
    @Html.AntiForgeryToken()

    <div>
        <label class="desc" for="title">Title</label>
        <div>
            <input id="title" name="title" type="text" class="field text
fn" value="@Model.Title" size="8" tabindex="1" autofocus>
        </div>
    </div>

    <div>
        <fieldset>
            <legend id="title5" class="desc">
                Status
            </legend>

            <div>
                <div>
                    <input id="status1" name="status"
type="radio" value="Open" tabindex="2" @(Model.Status == "Open" ? "checked" : "")>
                    <label class="choice"
for="status1">Open</label>
                </div>
                <div>
                    <input id="status2" name="status"
type="radio" value="In Progress" tabindex="3" @(Model.Status == "In Progress" ?
"checked" : "")>
                    <label class="choice"
for="status2">In Progress</label>
                </div>
            </div>
        </div>
    </div>
}

```

```

                                <input id="status3" name="status"
type="radio" value="Finished" tabindex="4" @(Model.Status == "Finished" ? "checked" :
"" )>
                                <label class="choice"
for="status3">Finished</label>
                                </div>
                                </div>
                                </fieldset>
                                </div>

                                <div>
                                <div>
                                <input class="button"
type="submit" value="Submit">
                                <a class="button cancel"
href="/">Cancel</a>
                                </div>
                                </div>

```

6. За да оформите стила на изгледите и оформлението ще ви са нужни два файла с разширение css, които да създадете в папка Content.

6.1 form-style.css файла съдържа следните свойства:

```

* {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}

body {
    padding: 20px 2%;
    font-family: sans-serif;
    font-size: 1.3em;
}

form header {
    margin: 0 0 20px 0;
}

form header div {
    font-size: 90%;
    color: #999;
}

form header h2 {
    margin: 0 0 5px 0;
}

form > div {
    clear: both;
    overflow: hidden;
    padding: 1px;
    margin: 0 0 10px 0;
}

form > div > fieldset > div > div {
    margin: 0 0 5px 0;
}

form > div > label,
legend {
    width: 25%;
    float: left;
}

```

```

        padding-right: 10px;
    }

    form > div > div,
    form > div > fieldset > div {
        width: 75%;
        float: right;
    }

    form > div > fieldset label {
        font-size: 90%;
    }

    fieldset {
        border: 0;
        padding: 0;
    }

    input[type=text],
    input[type=email],
    input[type=url],
    input[type=password],
    textarea {
        width: 100%;
        border: 1px solid #ccc;
        border-right-color: #eee;
        border-bottom-color: #eee;
        font-size: 1em;
    }

    input[type=text],
    input[type=email],
    input[type=url],
    input[type=password] {
        width: 50%;
    }

    input[type=text]:focus,
    input[type=email]:focus,
    input[type=url]:focus,
    input[type=password]:focus,
    textarea:focus {
        outline: 0;
        border-color: #4697e4;
    }

    .button {
        font-family: "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif;
        color: #fff;
        background-color: #0275d8;
        font-weight: 400;
        line-height: 1.25;
        text-align: center;
        white-space: nowrap;
        vertical-align: middle;
        user-select: none;
        border: 1px solid transparent;
        padding: .5rem 1rem;
        border-radius: .25rem;
        transition: border-color .15s ease-in-out;
        text-decoration: none;
        display: inline-block;
        font-size: .8em;
    }

```



```

.button:hover {
    background-color: #0284e9;
    border-color: black;
}

.cancel {
    cursor: default;
    color: #292b2c !important;
    background-color: #fff !important;
    border-color: #ccc !important;
}

.cancel:hover {
    color: #292b2c !important;
    background-color: #e6e6e6 !important;
    border-color: #adadad !important;
}

.cancel:focus, .cancel.focus {
    -webkit-box-shadow: 0 0 0 2px rgba(204, 204, 204, 0.5);
    box-shadow: 0 0 0 2px rgba(204, 204, 204, 0.5);
}

@media (max-width: 600px) {
    form > div {
        margin: 0 0 15px 0;
    }

    form > div > label,
    legend {
        width: 100%;
        float: none;
        margin: 0 0 5px 0;
    }

    form > div > div,
    form > div > fieldset > div {
        width: 100%;
        float: none;
    }

    input[type=text],
    input[type=email],
    input[type=url],
    input[type=password],
    textarea,
    select {
        width: 100%;
    }
}

@media (min-width: 1200px) {
    form > div > label,
    legend {
        text-align: right;
    }
}

```

6.2 index-style.css файла съдържа следните свойства:

```
@import url('https://fonts.googleapis.com/css?family=Lato');
```

```
* {
    margin: 0;
    padding: 0;
}

.container {
    margin: 15px auto;
    height: 80%;
    background: #fefefe;
}

.container.tasks {
    text-align: center;
}

.container a.button {
    margin-left: 3%;
}

.column {
    display: inline-block;
    border: 0.1em solid #4B4B4B;
    padding: 0.3em;
    vertical-align: top;
    width: 31%;
}

.column li {
    text-align: left;
    padding: 15px;
    border-radius: 8px;
    list-style: none;
    border: 0.15em solid #afafaf;
    box-shadow: 0 0 1px 1px #ccc;
    font-family: Lato, sans-serif;
    word-break: break-all;
    margin: 0.6em 0;
    background: #fafafa;
    transition: .15s background;
    font-size: 1.1em;
}

.column li:hover {
    background: #c4f5c4;
}

.header {
    white-space: nowrap;
    font-size: 1.3em;
    font-family: Lato, sans-serif;
    text-align: center;
    color: white;
    padding: 0.3em 4.5px;
    border-radius: 5px 5px 0 0;
}

.column.open {
    background: #7dbdbd;
}

.column.inProgress {
    background: #66ccff;
}

.column.finished {
```

```

        background: #c8c3e6;
    }

    .header.open {
        background: #30BFBF;
    }

    .header.inProgress {
        background: #00AAFF;
    }

    .header.finished {
        background: #8F7EE6;
    }

    a.button {
        font-family: "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif;
        color: #fff;
        background-color: #0275d8;
        font-weight: 400;
        line-height: 1.25;
        text-align: center;
        white-space: nowrap;
        vertical-align: middle;
        user-select: none;
        border: 1px solid transparent;
        padding: .5rem 1rem;
        border-radius: .25rem;
        transition: border-color .15s ease-in-out;
        text-decoration: none;
    }

    a.button:hover {
        background-color: #0284e9;
        border-color: black;
    }

    .column li:hover .icon {
        opacity: 1;
    }

    .icon {
        opacity: 0;
        text-decoration: none;
        color: black;
        transition: .15s all;
    }

    .icon:hover {
        color: #8a8a8a;
    }

    .edit:before {
        content: "✎";
    }

```

7. Давам ви кода за TaskController-а само за изглед index и create, останалите си ги създайте сами.

```

using System.Linq;
using System.Web.Mvc;
using TeisterMask.Models;

```

## ИНТЕРНЕТ ПРОГРАМИРАНЕ

```
namespace TeisterMask.Controllers
{
    [ValidateInput(false)]
    public class TaskController : Controller
    {
        [HttpGet]
        [Route("")]
        public ActionResult Index()
        {
            using (var db = new TeisterMaskDbContext())
            {
                return View(db.Tasks.ToList());
            }
        }

        [HttpGet]
        [Route("create")]
        public ActionResult Create()
        {
            return View();
        }

        [HttpPost]
        [Route("create")]
        [ValidateAntiForgeryToken]
        public ActionResult Create(Task task)
        {
            if (!ModelState.IsValid)
                return RedirectToAction("Index");

            using (var db = new TeisterMaskDbContext())
            {
                db.Tasks.Add(task);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
        }
    }
}

// допишете и за останалите изгледи логиката
```