View:

@using project.Models

@model project.Models.RichTextEditorViewModel

@{

ViewBag.Title = "Text";

}

<h2>@ViewBag.Title.</h2>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css" />

<div class = "row" >

<div class = "col-md-12" >

<section id = "loginForm" >

@using (Html.BeginForm("RichTextEditor", "Home", new{ReturnUrl = ViewBag.ReturnUrl}, FormMethod.Post, new

{

@class = "form-horizontal",

role = "form"

}))

{

@Html.AntiForgeryToken()

<div class = "form-group" > <label class = "col-md-4 control-label" > Message </label> <div class = "col-md-8" > <div class = "input-group" > @Html.TextAreaFor(m => m.Message, new

{

rows = "20",

style = "resize:none; width:400px;",

placeholder = Html.DisplayNameFor(m => m.Message),

@class = "form-control input-lg textarea-editor"

})

</div>

</div>

</div>

} </section> </div> </div>

@section Scripts {

<script type = "text/javascript" src = "http://cdnjs.cloudflare.com/ajax/libs/summernote/0.8.2/summernote.js" >

</script>

@Scripts.Render("~/bundles/script-custom-editor")

<link rel = "stylesheet" href = "http://cdnjs.cloudflare.com/ajax/libs/summernote/0.8.2/summernote.css">

}

Model:

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using Microsoft.AspNet.Identity;

using Microsoft.Owin.Security;

using System.Web.Mvc;

namespace project.Models

{

public class RichTextEditorViewModel

{

[AllowHtml]

[Display(Name = "Message")]

public string Message

{

get;

set;

}

}

}

Script:

$(document).ready(function () {

// Initialize Editor

$('.textarea-editor').summernote({

height: 300, // set editor height

minHeight: null, // set minimum height of editor

maxHeight: null, // set maximum height of editor

focus: true // set focus to editable area after initializing summernote

});

});

Controller:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using project.Models;

using System.Net;

using System.Net.Mail;

using System.Threading.Tasks;

using System.Web.UI.WebControls;

using System.IO;

namespace project.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

return View();

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<ActionResult> Contact(EmailFormModel model)

{

if (ModelState.IsValid)

{

var body = "<p>Email From: {0} ({1})</p><p>Message:</p><p>{2}</p>";

var message = new MailMessage();

message.To.Add(new MailAddress("zhua0056@monash.edu.com")); // replace with valid value

message.From = new MailAddress("zhua0056@monash.edu.com"); // replace with valid value

message.Subject = "Your email subject";

message.Body = string.Format(body, model.FromName, model.FromEmail, model.Message);

message.IsBodyHtml = true;

if (model.Upload != null && model.Upload.ContentLength > 0)

{

message.Attachments.Add(new

Attachment(model.Upload.InputStream,

System.IO.Path.GetFileName(model.Upload.FileName)));

}

using (var smtp = new SmtpClient())

{

var credential = new NetworkCredential

{

//UserName = "user@outlook.com", // replace with valid value

//Password = "password" // replace with valid value

};

smtp.Credentials = credential;

//smtp.Host = "smtp-mail.outlook.com";

smtp.Host = "smtp.monash.edu.au";

//smtp.Port = 587;

//smtp.EnableSsl = true;

await smtp.SendMailAsync(message);

return RedirectToAction("Sent");

}

}

return View(model);

}

public ActionResult Sent()

{

return View();

}

public ActionResult RichTextEditor()

{

return View();

}

}

}