
Overview of SAT Requirements

This SAT system was designed by Abbey Outain and Andrew Allen. The SAT provides scheduling tools for Administrators and Schedulers as well as enrollment tools for Students, Administrators and Schedulers.

The requirements of the SAT system include:

- Must be mobile responsive.
- 2-tier plus design, including a database for the provided database schema.
- Convert an appropriate multi page template.
- Must have built in internal security to lock certain functionality for non admins.
- Utilize source control through a collaboration on GitHub.

Admin Capabilities Include

1. CRUD functionality for Courses, Students, Enrollment, Scheduled Classes, and Student Statuses.
2. View all details about each mentioned list above.
3. Edit all data within each mentioned list above.

Scheduler Capabilities Include

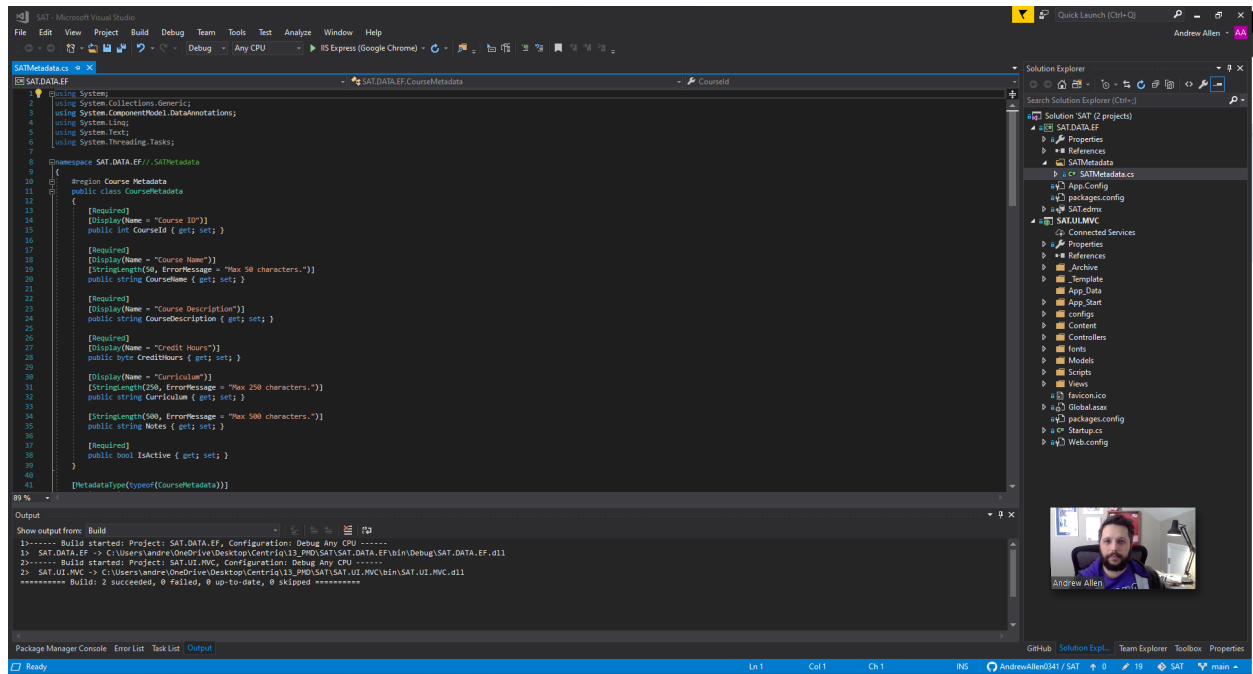
1. CRUD functionality for Enrollments and Scheduled Classes.
2. The ability to view all students with accompanying details.

Anonymous User Capabilities Include

1. View the website home page and submit contact form.
2. Unable to manipulate any data in Courses, Students, Enrollment, Scheduled Classes, and Student Statuses.

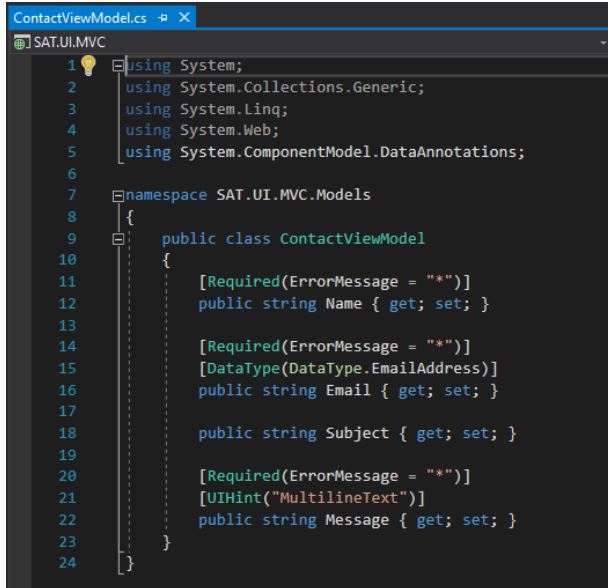
Metadata

Data that provides information about other data. Metadata summarizes basic information about data, making finding and working with particular instances of data easier. Metadata for this project was created manually to be more accurate, custom rules were also created to add additional functionality to our basic information.



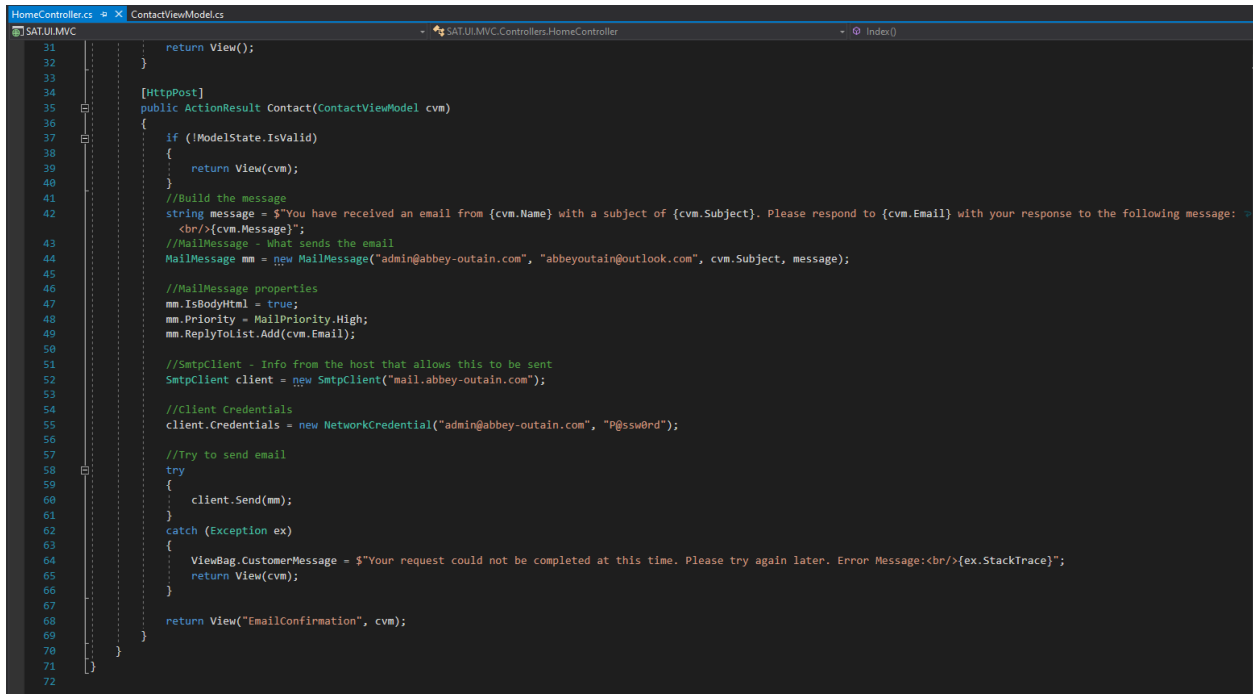
Contact Form

Created the ContactViewModel by right-clicking on the Model folder and adding a class.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.ComponentModel.DataAnnotations;
6
7 namespace SAT.UI.MVC.Models
8 {
9     public class ContactViewModel
10     {
11         [Required(ErrorMessage = "**")]
12         public string Name { get; set; }
13
14         [Required(ErrorMessage = "**")]
15         [DataType(DataType.EmailAddress)]
16         public string Email { get; set; }
17
18         public string Subject { get; set; }
19
20         [Required(ErrorMessage = "**")]
21         [UIHint("MultilineText")]
22         public string Message { get; set; }
23     }
24 }
```

Created Post for the Contact action in HomeController.



```
31 return View();
32 }
33
34 [HttpPost]
35 public ActionResult Contact(ContactViewModel cvm)
36 {
37     if (!ModelState.IsValid)
38     {
39         return View(cvm);
40     }
41
42     //Build the message
43     string message = $"You have received an email from {cvm.Name} with a subject of {cvm.Subject}. Please respond to {cvm.Email} with your response to the following message: {cvm.Message}";
44     //MailMessage - What sends the email
45     MailMessage mm = new MailMessage("admin@abbey-outain.com", "abbeyoutain@outlook.com", cvm.Subject, message);
46
47     //MailMessage properties
48     mm.IsBodyHtml = true;
49     mm.Priority = MailPriority.High;
50     mm.ReplyToList.Add(cvm.Email);
51
52     //SmtpClient - Info from the host that allows this to be sent
53     SmtpClient client = new SmtpClient("mail.abbey-outain.com");
54
55     //Client Credentials
56     client.Credentials = new NetworkCredential("admin@abbey-outain.com", "P@ssw0rd");
57
58     //Try to send email
59     try
60     {
61         client.Send(mm);
62     }
63     catch (Exception ex)
64     {
65         ViewBag.CustomerMessage = $"Your request could not be completed at this time. Please try again later. Error Message:<br/>{ex.StackTrace}";
66         return View(cvm);
67     }
68
69     return View("EmailConfirmation", cvm);
70 }
71
72 }
```

Created AppSecretKeys.config in the configs folder to store this sensitive data.

```
AppSecretKeys.config  HomeController.cs  ContactViewModel.cs
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <appSettings>
4   <!--This file was added to protect sensitive data from being committed to Source Control-->
5
6   <!--Replace yourdomain.com below with your actual domain and extension-->
7   <add key="EmailClient" value="mail.abbey-outain.com"/>
8
9   <!--Replace mail@yourdomain.com below with the email user created for your hosting account-->
10  <add key="EmailUser" value="admin@abbey-outain.com"/>
11
12  <!--Replace Password below with your email user's password-->
13  <add key="EmailPass" value="P@ssw0rd"/>
14
15  <!--Replace mail@domain.com below with your personal email-->
16  <add key="EmailTo" value="abbeyoutain@outlook.com"/>
17
18 </appSettings>
```

In the Web.config, added file attribute to the appSettings tag in order to link to AppSecretKeys.

```
7 <appSettings file="configs\AppSecretKeys.config">
8   <add key="owin:AppStartup" value="SAT.UI.MVC.Startup,SAT.UI.MVC" />
9   <add key="owin:AppStartup" value="SAT.UI.MVC.Startup,SAT.UI.MVC" />
10  <add key="webpages:Version" value="3.0.0.0" />
11  <add key="webpages:Enabled" value="false" />
12  <add key="ClientValidationEnabled" value="true" />
13  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
14 </appSettings>
```

Generated and styled the new Contact View.

```
1 @model SAT.UI.MVC.Models.ContactViewModel
2
3 ViewBag.Title = "Contact";
4
5
6
7 <section class="section video" data-section="section5">
8     <div class="section-heading">
9         <h2>Contact Us</h2>
10    </div>
11    <div class="container">
12        <div class="row">
13            <div class="col-md-6 align-self-center">
14                <div class="left-content">
15                    <span>The KC Culinary Academy is for you</span>
16                    <h4>Contact us to learn <em>about KC Culinary Academy</em></h4>
17                    <p>
18                        For us to better serve a potential student please fill out this form to the fullest including a brief message on why you are pursuing a
19                        degree in Culinary Arts or Pastry Arts.
20                        <br><br>Thank you for your interest and you will receive an email response within 24-48 hours.
21                    </p>
22                    <div class="main-button"><a rel="nofollow" href="https://www.youtube.com/watch?v=jd2m1Iie91g" target="_blank">External URL</a></div>
23                </div>
24            <div class="col-md-6">
25                @using (Html.BeginForm())
26                {
27                    @Html.AntiForgeryToken()
28
29                    <div class="form-horizontal">
30                        <h4><span>Contact</span> Us</h4>
31
32                        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
33                        <div class="form-group">
34                            @Html.LabelFor(model => model.Name, htmlAttributes: new { @class = "control-label col-md-2" })
35                            <div class="col-md-10">
36                                @Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "form-control" } })
37                                @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })
38                            </div>
39                        </div>
40
41                        <div class="form-group">
42                            @Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })
43                            <div class="col-md-10">
44                                @Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })
45                                @Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })
46                            </div>
47                        </div>
48
49                        <div class="form-group">
50                            @Html.LabelFor(model => model.Subject, htmlAttributes: new { @class = "control-label col-md-2" })
51                            <div class="col-md-10">
52                                @Html.EditorFor(model => model.Subject, new { htmlAttributes = new { @class = "form-control" } })
53                                @Html.ValidationMessageFor(model => model.Subject, "", new { @class = "text-danger" })
54                            </div>
55                        </div>
56
57                        <div class="form-group">
58                            @Html.LabelFor(model => model.Message, htmlAttributes: new { @class = "control-label col-md-2" })
59                            <div class="col-md-10">
60                                @Html.EditorFor(model => model.Message, new { htmlAttributes = new { @class = "form-control" } })
61                                @Html.ValidationMessageFor(model => model.Message, "", new { @class = "text-danger" })
62                            </div>
63                        </div>
64
65                        <div class="form-group">
66                            <div class="col-md-offset-2 col-md-10">
67                                <input type="submit" value="Send" class="btn btn-default" />
68                            </div>
69                        </div>
70                    </div>
71                }
72            </div>
73        </div>
74    </div>
75    <br />
76    <br />
77    <br />
78    <br />
79    <br />
80
81 </section>
82
```

Generated and styled the EmailConfirmation View.

```
EmailConfirmation.cshtml # X Contact.cshtml Web.config AppSecretKeys.config HomeController
1 @model SAT.UI.MVC.Models.ContactViewModel
2
3
4 ViewBag.Title = "EmailConfirmation";
5
6
7 <h2>EmailConfirmation</h2>
8
9 <div>
10     <h4>ContactViewModel</h4>
11     <hr />
12     <dl class="dl-horizontal">
13         <dt>
14             @Html.DisplayNameFor(model => model.Name)
15         </dt>
16         <dd>
17             @Html.DisplayFor(model => model.Name)
18         </dd>
19     </dl>
20     <dt>
21         @Html.DisplayNameFor(model => model.Email)
22     </dt>
23     <dd>
24         @Html.DisplayFor(model => model.Email)
25     </dd>
26     <dt>
27         @Html.DisplayNameFor(model => model.Subject)
28     </dt>
29     <dd>
30         @Html.DisplayFor(model => model.Subject)
31     </dd>
32     <dt>
33         @Html.DisplayNameFor(model => model.Message)
34     </dt>
35     <dd>
36         @Html.DisplayFor(model => model.Message)
37     </dd>
38 </dl>
39 </div>
40 <p>
41     @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
42     @Html.ActionLink("Back to List", "Index")
43 </p>
44
45
46
47
48
49
50
51
```

Final Contact Page:

Contact Us

THE KC CULINARY ACADEMY IS FOR YOU

Contact us to learn about KC Culinary Academy

For us to better serve a potential student please fill out this form to the fullest including a brief message on why you are pursuing a degree in Culinary Arts or Pastry Arts.

Thank you for your interest and you will receive an email response within 24-48 hours.

EXTERNAL URL

Name

Email

Subject

Message

SEND