

SAT TOOL DOCUMENTATION

REQUIREMENTS

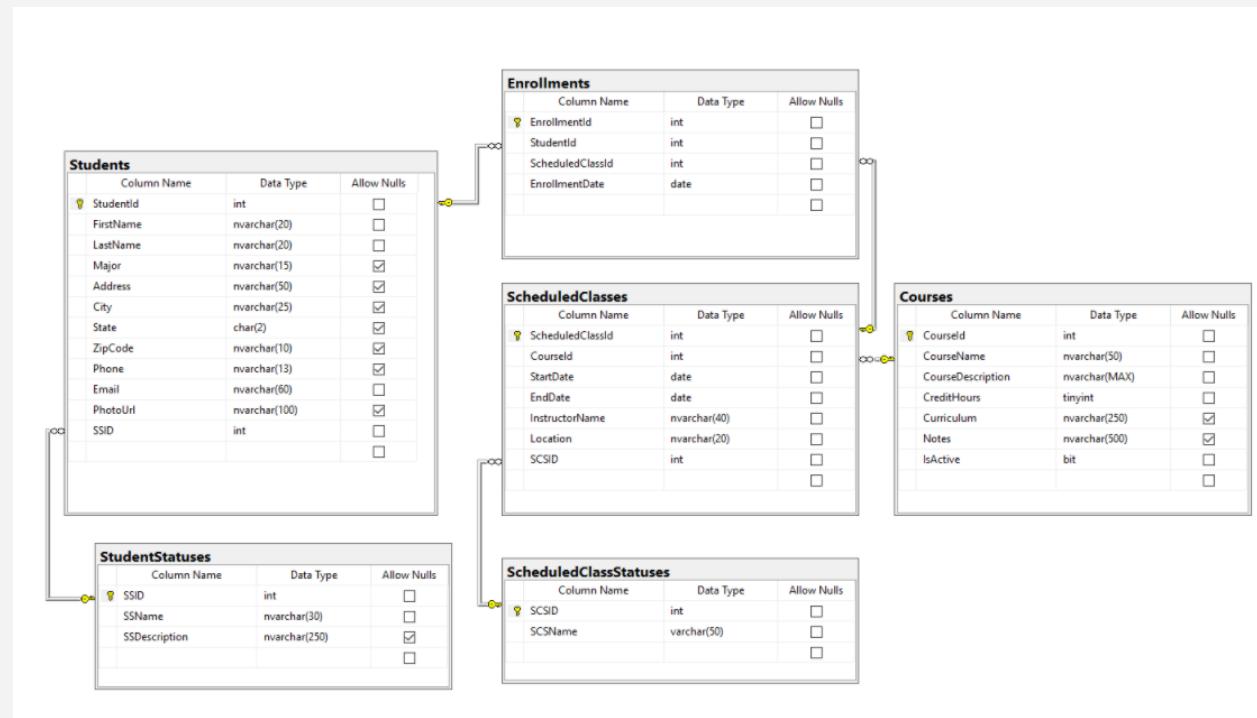
Summary:

We were tasked to build a Scheduling Administration Tool (SAT) to aid staff with enrolling students, managing courses, and managing class scheduling at an institution of higher learning.

An Identity Matrix (Right) was provided to instruct us on how to scaffold out controllers, views, and which user roles can see those views.

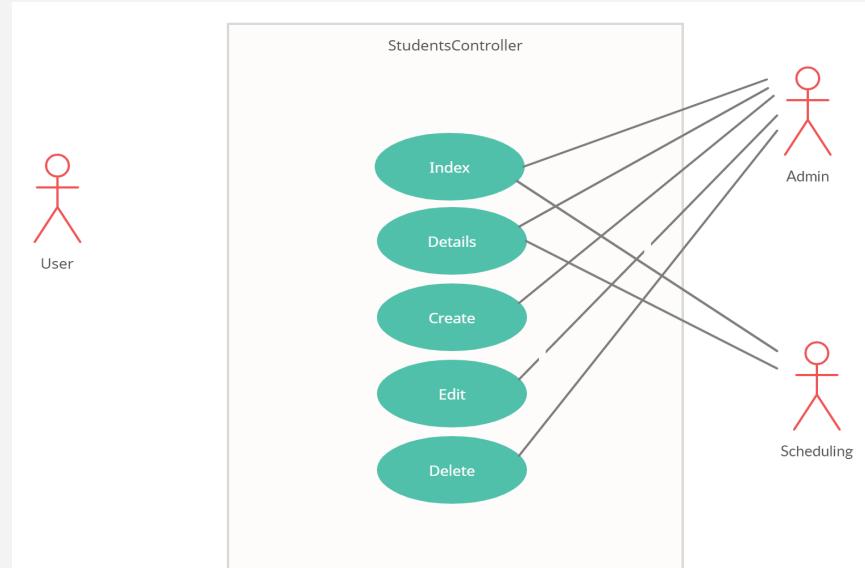
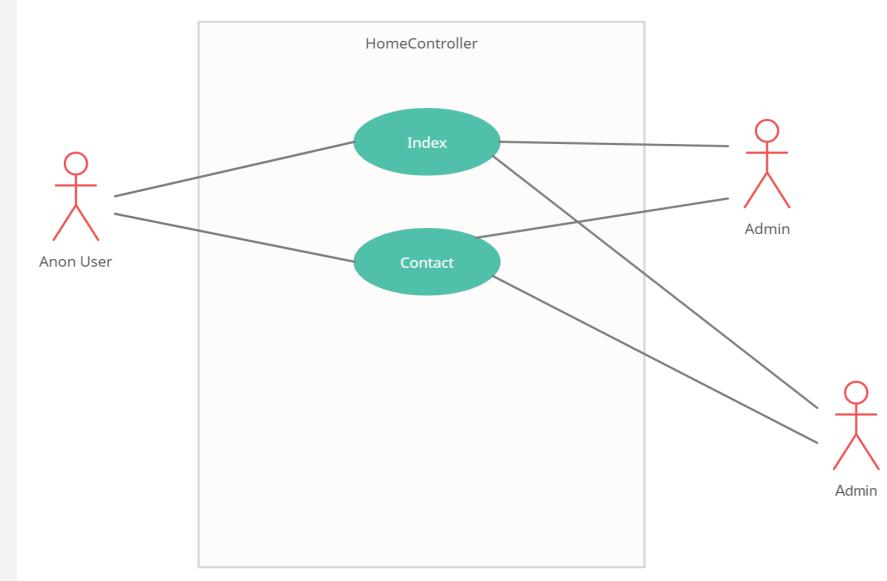
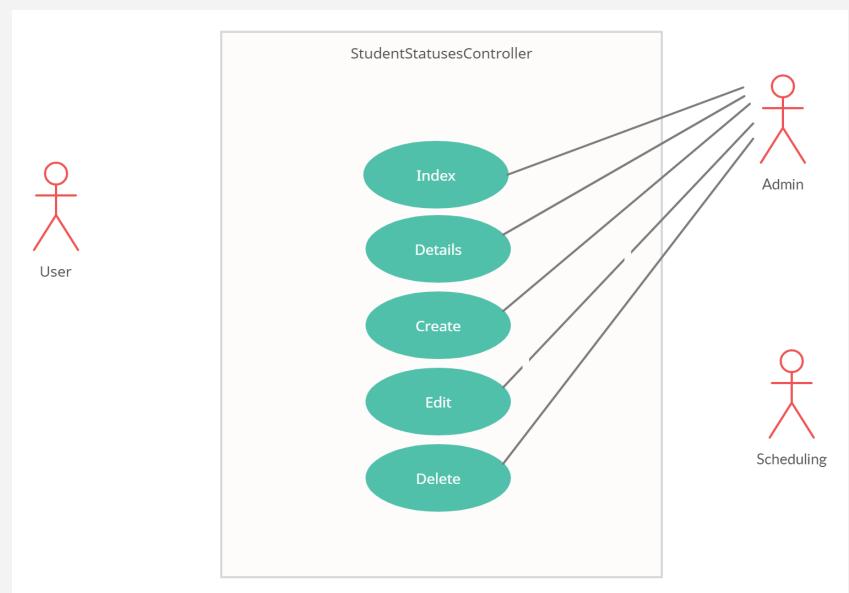
	Index	Contact					
HomeController	x	x					
Anonymous							
Scheduling	x	x					
Admin	x	x					
Courses	Active	Retired	Detail	Create	Edit	Delete	
Anonymous							
Scheduling							
Admin	x	x	x	x	x	x	x
Enrollments	Index	Details	Create	Edit	Delete		
Anonymous							
Scheduling	x	x	x	x	x	x	
Admin	x	x	x	x	x	x	x
Scheduled Classes	Index	Details	Create	Edit	Delete		
Anonymous							
Scheduling	x	x	x	x	x		
Admin	x	x	x	x	x	x	x
Students	Index	Details	Create	Edit	Delete		
Anonymous							
Scheduling	x	x					
Admin	x	x	x	x	x	x	x
Student Statuses	Index	Details	Create	Edit	Delete		
Anonymous							
Scheduling							
Admin	x	x	x	x	x	x	x

We were also tasked to create and use a relational database based off of the database schema provided to us (Below).

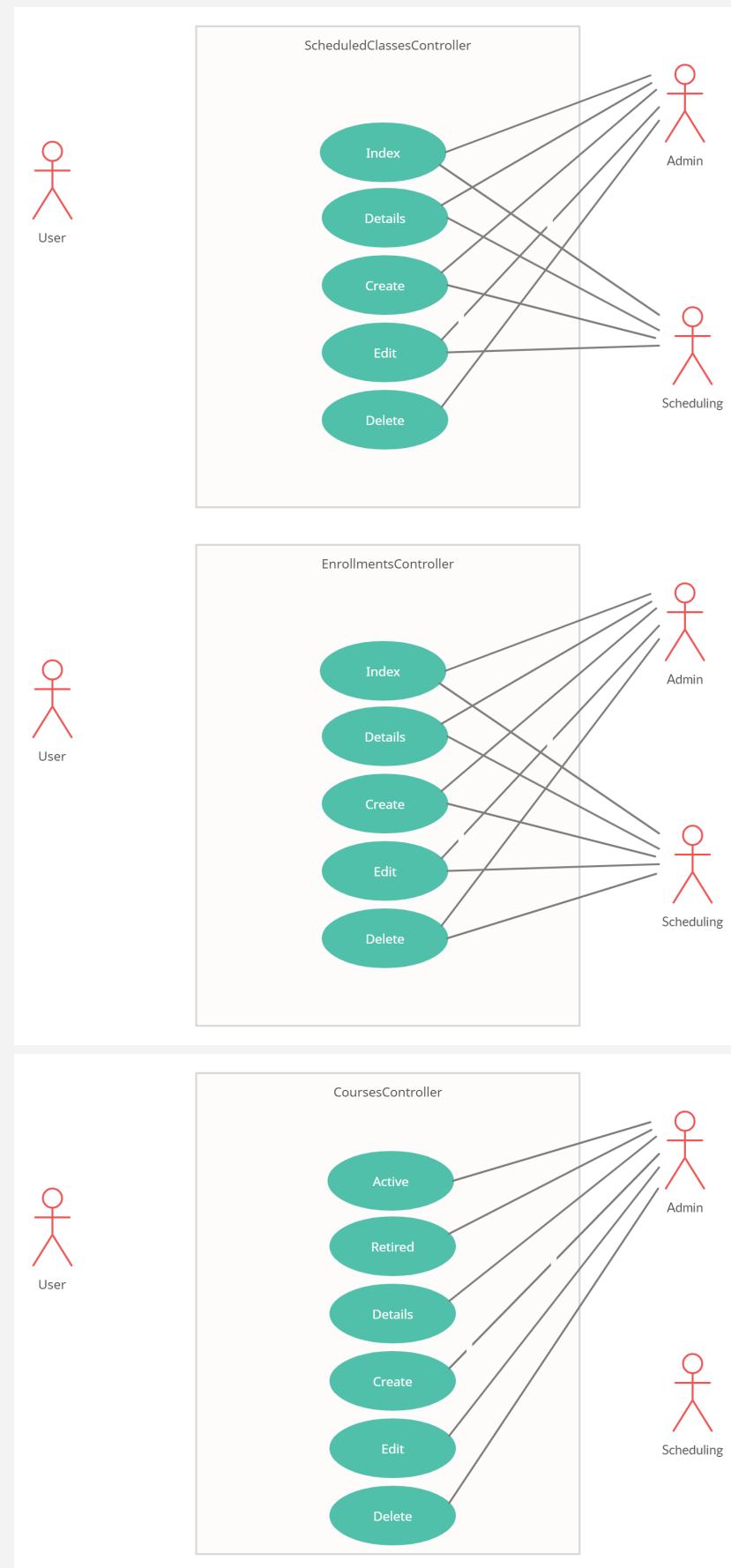


SAT TOOL DOCUMENTATION

USE CASE DIAGRAMS



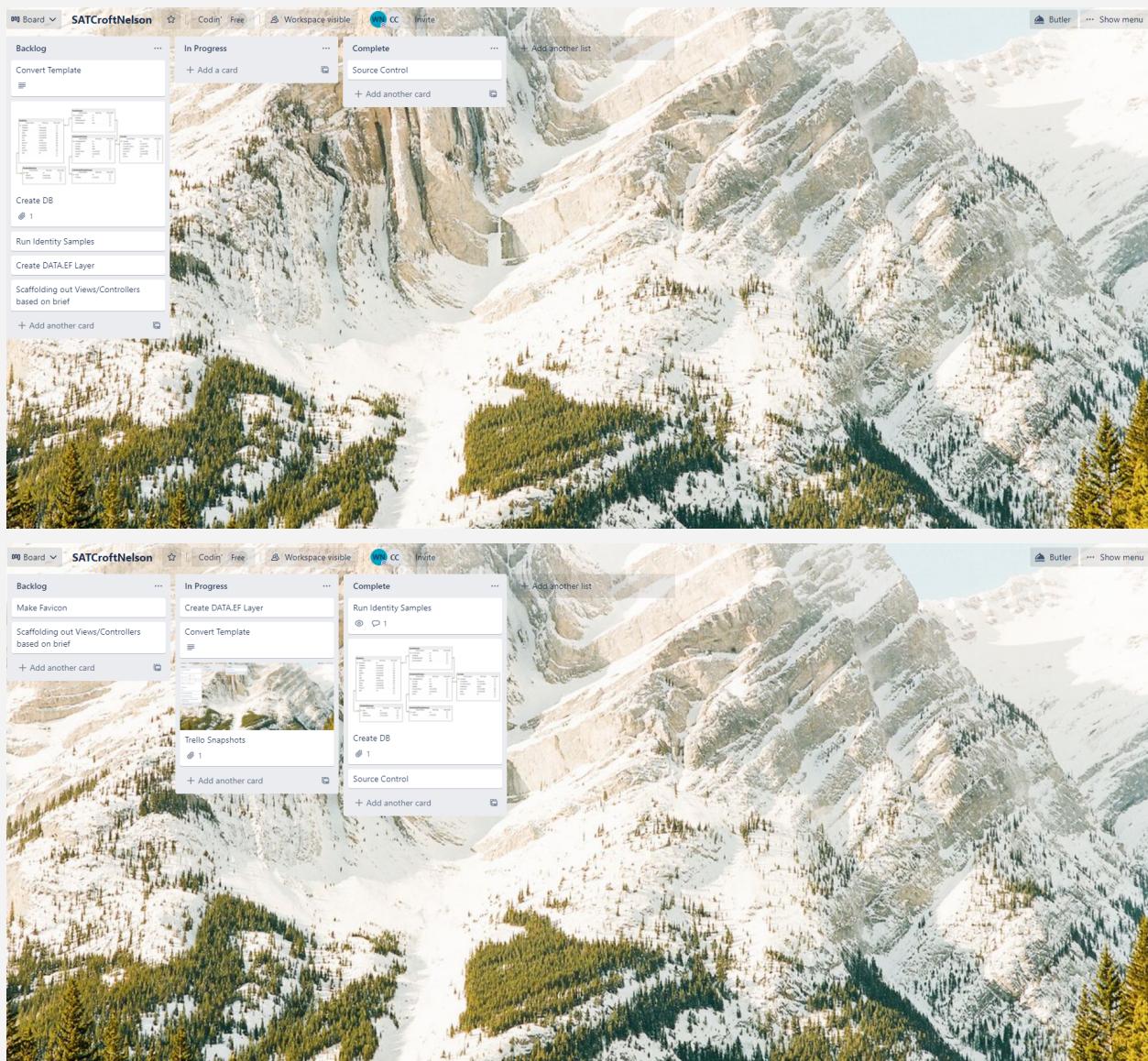
SAT TOOL DOCUMENTATION



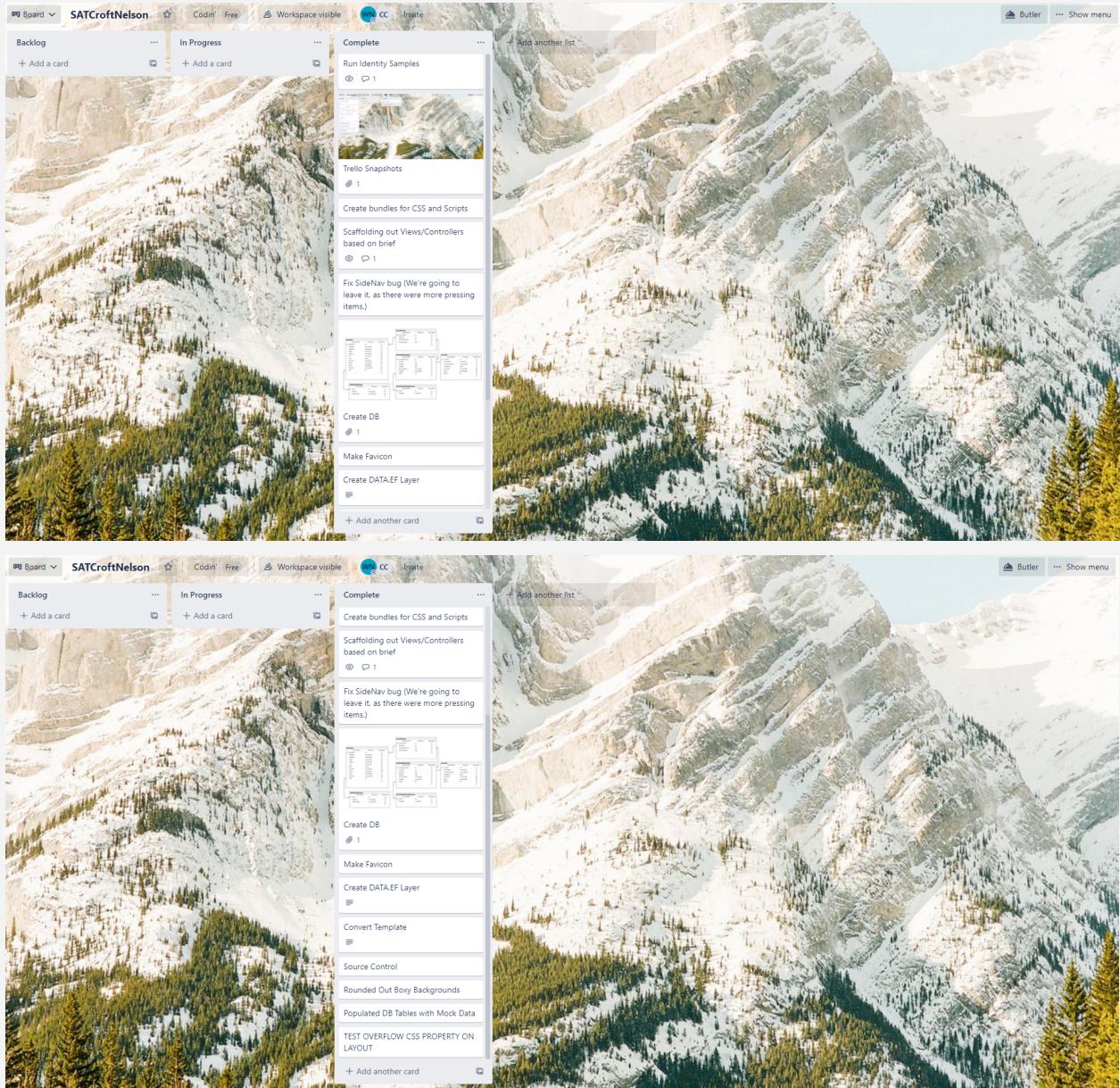
SAT TOOL DOCUMENTATION

PROJECT MANAGEMENT

We utilized a kanban-style tool named Trello to track our tasks and make our development process more efficient. Below are screenshots depicting our Trello board through multiple stages of project completion.



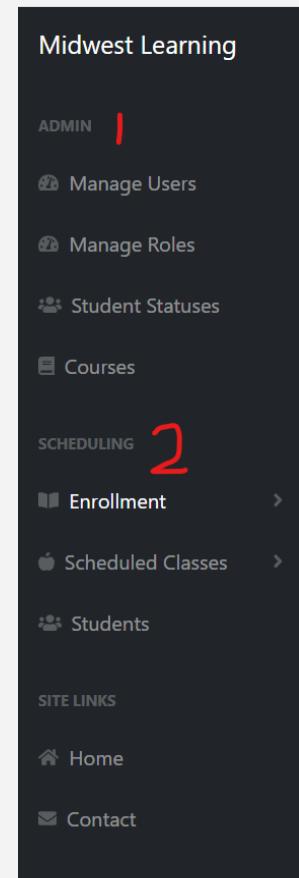
SAT TOOL DOCUMENTATION



SAT TOOL DOCUMENTATION

MEETING PROJECT REQUIREMENTS

```
<div id="layoutSidenav">
  <div id="layoutSidenav_nav">
    <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
      <div class="sb-sidenav-menu">
        <div class="nav">
          <div class="nav-item">
            <div class="nav-link">
              <a href="#">Admin</a>
            </div>
            <div class="nav-link">
              <a href="#">Manage Users</a>
            </div>
            <div class="nav-link">
              <a href="#">Manage Roles</a>
            </div>
            <div class="nav-link">
              <a href="#">Student Statuses</a>
            </div>
            <div class="nav-link">
              <a href="#">Courses</a>
            </div>
          </div>
        </div>
      </div>
    </nav>
  </div>
</div>
```



To ensure only users who fall under a certain role have access to certain functionality within our web application, we use authentication and authorization. These if statements within the View (.cshtml file) lock down what the users in certain roles can SEE. #1 in the figure above is displaying the content within the block to anyone who is an Admin. #2 in the figure above is allowing Admin or Scheduling to see that piece of the View.

The figure to the right shows what content is locked down by each if statement.

```
[Authorize(Roles = "Admin,Scheduling")]
public class ScheduledClassesController : Controller
{}
```

Above is locking down certain functionality within our web application. Because it is applied to the Controller and not a View, this is instead changing what a user in a certain role can DO. If a user of an unauthorized role tries to access an authorized piece of functionality, they are rerouted to the login page.

SAT TOOL DOCUMENTATION

```
<div id="studentStatusGrid" class="row"> ●  
  @foreach (var item in Model) ●  
  {  
    ● <div class="card col-md-3 col-sm-6 col-xs-6 studentStatus">  
      | <h4 class="text-center">  
      |   <a href="@Url.Action("Details", "StudentStatus", new { id = item.SSID })" class="text-center">@item.SSName</a>  
      | </h4>  
      | <div class="statusDescription">  
      |   <p>@item.SSDescription</p>  
      | </div>  
    </div>  
  }  
</div>
```

Green: Added custom CSS properties to an ID of studentStatusGrid, and added a bootstrap class of row.

Red: For each item (in this case, objects of type StudentStatus), render the contents of the foreach body.

Blue: Here, we style the individual tiles. We use the bootstrap class of card to give the div a card style, and then the col classes to format their positions. A custom class of studentStatus adds some margin on the bottom of each card.

Yellow: We center the Student Status name on the card with a bootstrap class of text-center. The @Url.Action takes us to the Details view of the StudentStatus displayed in the card. The anonymous property renders a dynamic route value. The class of statusDescription helps to style the div, which has a paragraph tag nested inside of it - which holds a dynamic Student Status description.

SAT TOOL DOCUMENTATION

ADDING DATA THROUGH THE USER INTERFACE

Active Courses					
Course Name		Course Description	Credit Hours	Curriculum	Notes
					Is Active
Intro to Data	Learn all about data.	5	Datatypes, Data Management, Databases	A public, free study group for Intro to Data meets on Wednesday night in the Student Union.	<input type="checkbox"/> Edit Details Delete
Advanced Python and Technical Prep	Learn upper-level python techniques and get ready for technical interviews.	6	Python Syntax, Advanced Techniques, Mock Interviews	[N/A]	<input type="checkbox"/> Edit Details Delete
Discrete Structures	Teaches the student logic and logical operations. Topics include logic gates, breaking down logic, and implementing logic into everyday life.	5	Logic	A very fun course!	<input type="checkbox"/> Edit Details Delete

Click the blue “Create New” button (1) at the top of the page to add a record.

Create

Course

Course Name

Course Description

Credit Hours

Curriculum

Notes

Is Active

[Create](#)

[Home](#)

Enter all required fields, and then click the blue “Create” button (1) at the bottom left of the creation form.