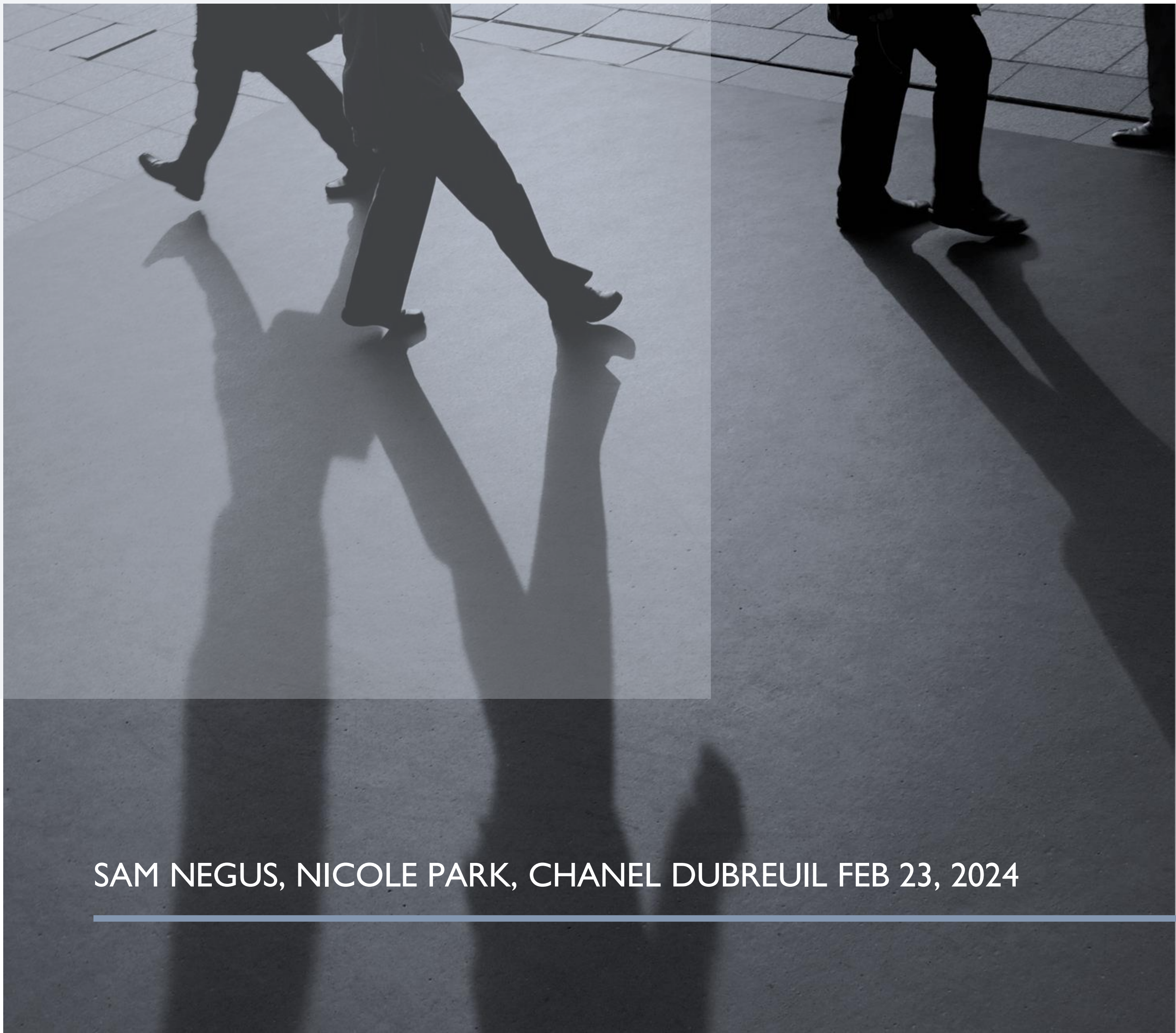
SCHEDULING ADMINISTRATION TOOL

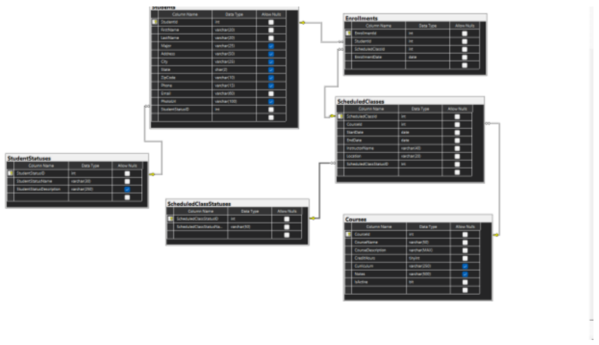
EASILY ENROLL STUDENTS IN YOUR CLASSES



SAM NEGUS, NICOLE PARK, CHANEL DUBREUIL FEB 23, 2024

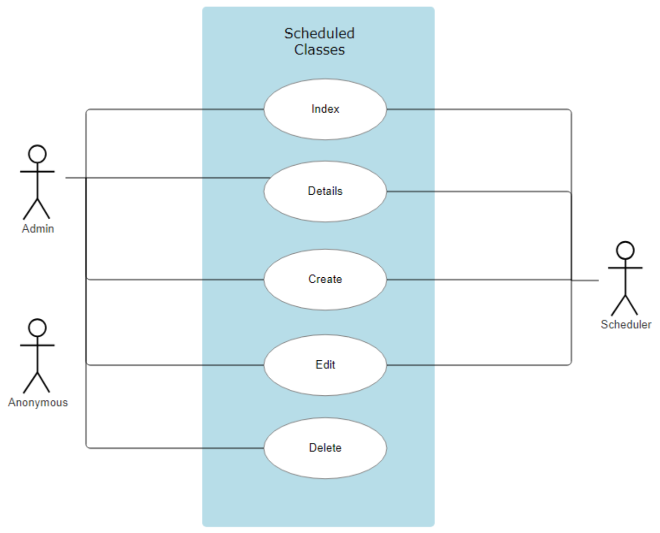
PROJECT OVERVIEW

We were tasked as a group with creating a multi-page, user authenticated web application to be used by a university to easily enroll students in classes, add new students to the university, create new courses, and maintain course and student statuses. The client asked that the program have different levels of authorization for roles such as Admin, Scheduler and Anonymous users.



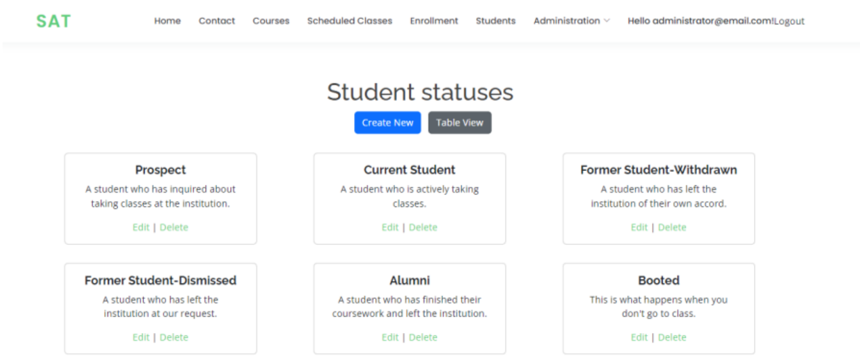
PROJECT REQUIREMENTS

- ❖ Create a custom, relational database
- ❖ Convert a template to MVC architecture
- ❖ Implement Identity roles for different levels of authorization
  - ➢ Admin
    - ▪ Ability to Create, Read, Update and Delete Students
    - ▪ Ability to Create, Read, Update and Delete Courses
    - ▪ Ability to Create, Read, Update and Delete Scheduled Classes
    - ▪ Ability to Create, Read, Update and Delete Enrollment
  - ➢ Scheduler
    - ▪ Ability to View Students
    - ▪ Ability to Create, Read and Update Scheduled Classes
    - ▪ Ability to Create, Read, Update and Delete Enrollment
  - ➢ Anonymous
    - ▪ Ability to View Home Page Login and Contact
- ❖ Customize Views and styling to be consistent



(Optional)

- ❖ Create custom properties to accurately summarize information
- ❖ Implement image upload functionality
- ❖ Create a tiled view for Student Status table



Student statuses

Prospect
A student who has inquired about taking classes at the institution.
Edit | Delete

Current Student
A student who is actively taking classes.
Edit | Delete

Former Student-Withdrawn
A student who has left the institution of their own accord.
Edit | Delete

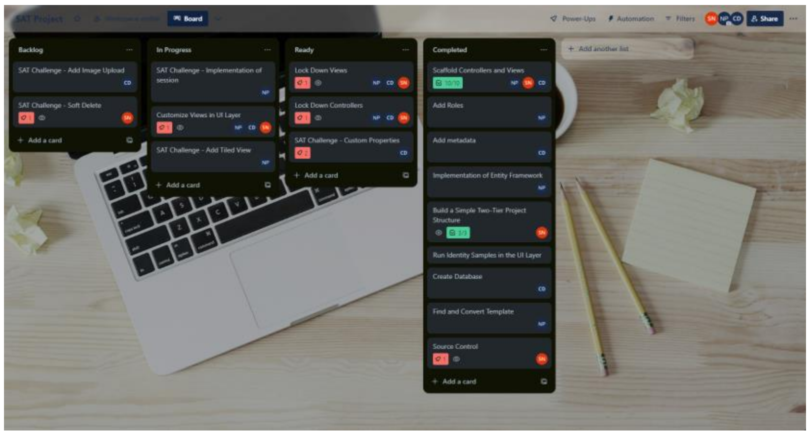
Former Student-Dismissed
A student who has left the institution at our request.
Edit | Delete

Alumni
A student who has finished their coursework and left the institution.
Edit | Delete

Booted
This is what happens when you don't go to class.
Edit | Delete

TECHNOLOGY USED

- SQL Server
- Visual Studio 2022
- GitHub
- Trello
- HTML
- C#
- CSS
- JavaScript



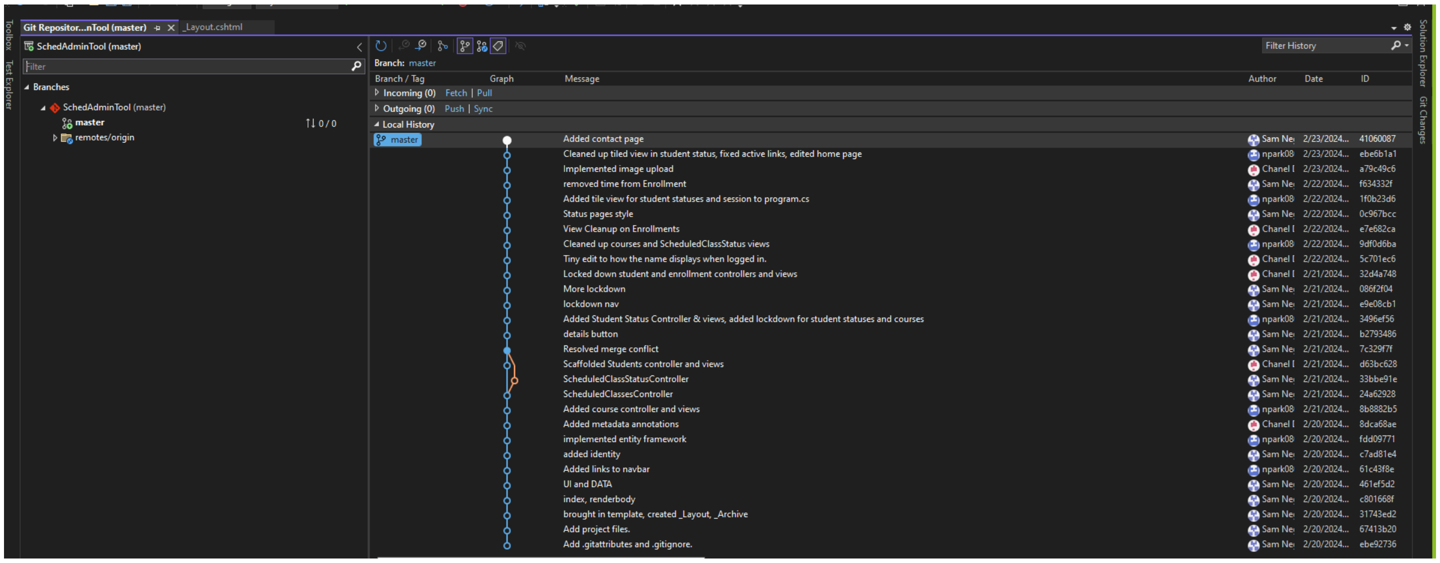
SAT Project | Board | Power-Ups | Automation | Filters | SN NP CD | Share

Backlog
- SAT Challenge - Add Image Upload (CD)
- SAT Challenge - Soft Delete (SN)

In Progress
- SAT Challenge - Implementation of session (NP)
- Customize Views in UI Layer (NP, CD, SN)
- SAT Challenge - Add Tiled View (NP)

Ready
- Lock Down Views (NP, CD, SN)
- Lock Down Controllers (NP, CD, SN)
- SAT Challenge - Custom Properties (CD)

Completed
- Scaffold Controllers and Views (10/10) (NP, SN, CD)
- Add Roles (NP)
- Add metadata (CD)
- Implementation of Entity Framework (NP)
- Build a Simple Two-Tier Project Structure (3/3) (SN)
- Run Identity Samples in the UI Layer
- Create Database (CD)
- Find and Convert Template (NP)
- Source Control (SN)



Git Repositor...nTool (master) | _Layout.cshtml

SchedAdminTool (master)

Branches
- SchedAdminTool (master)
  - master ↑↓ 0 / 0
  - remotes/origin

Branch: master

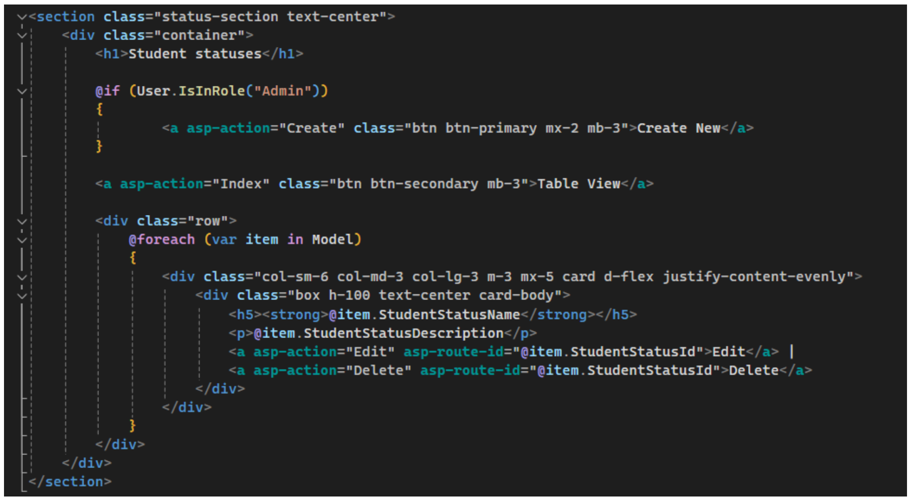
Incoming (0) Fetch | Pull
Outgoing (0) Push | Sync
Local History

Branch / Tag	Message	Author	Date	ID
master	Added contact page	Sam Ne	2/23/2024...	41060087
	Cleaned up tiled view in student status, fixed active links, edited home page	npark08	2/23/2024...	ebe6b1a1
	Implemented image upload	Chanel I	2/23/2024...	a79c49c6
	removed time from Enrollment	Sam Ne	2/22/2024...	f634332f
	Added tile view for student statuses and session to program.cs	npark08	2/22/2024...	1f0b23d6
	Status pages style	Sam Ne	2/22/2024...	0c967bcc
	View Cleanup on Enrollments	Chanel I	2/22/2024...	e7e682ca
	Cleaned up courses and ScheduledClassStatus views	npark08	2/22/2024...	9df0d6ba
	Tiny edit to how the name displays when logged in.	Chanel I	2/22/2024...	5c701ec6
	Locked down student and enrollment controllers and views	Chanel I	2/21/2024...	32d4a748
	More lockdown	Sam Ne	2/21/2024...	086f2f04
	lockdown nav	Sam Ne	2/21/2024...	e9e08cb1
	Added Student Status Controller & views, added lockdown for student statuses and courses	npark08	2/21/2024...	3496ef56
	details button	Sam Ne	2/21/2024...	b2793486
	Resolved merge conflict	Sam Ne	2/21/2024...	7c329f7f
	Scaffolded Students controller and views	Chanel I	2/21/2024...	d63bc628
	ScheduledClassStatusController	Sam Ne	2/21/2024...	33bbe91e
	ScheduledClassesController	Sam Ne	2/21/2024...	24a62928
	Added course controller and views	npark08	2/21/2024...	8b8882b5
	Added metadata annotations	Chanel I	2/20/2024...	8dca68ae
	implemented entity framework	npark08	2/20/2024...	fdd09771
	added identity	Sam Ne	2/20/2024...	c7ad81e4
	Added links to navbar	npark08	2/20/2024...	61c43f8e
	UI and DATA	Sam Ne	2/20/2024...	461ef5d2
	index, renderbody	Sam Ne	2/20/2024...	c801668f
	brought in template, created _Layout, _Archive	Sam Ne	2/20/2024...	31743ed2
	Add project files.	Sam Ne	2/20/2024...	67413b20
	Add .gitattributes and .gitignore.	Sam Ne	2/20/2024...	ebe92736

CODE SNIPPETS

Creating a Tiled View:

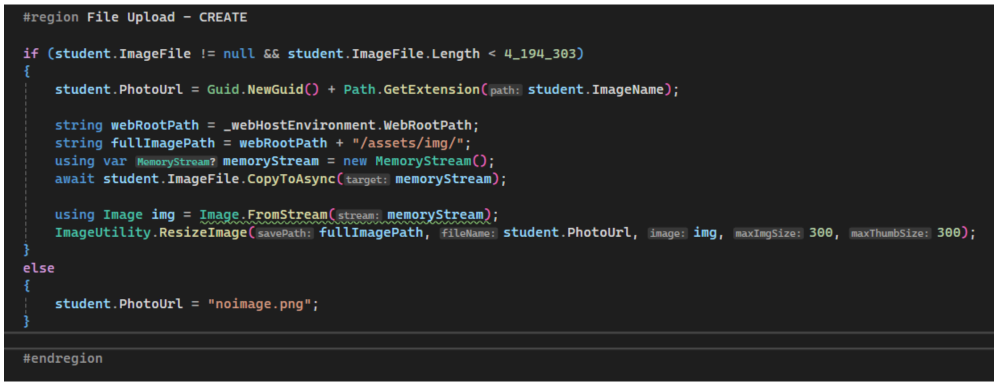
Here we created a Tiled View of our Student Statuses so the user can switch between a table and tile view with the click of a button!



```
<section class="status-section text-center">
    <div class="container">
        <h1>Student statuses</h1>

        @if (User.IsInRole("Admin"))
        {
                <a asp-action="Create" class="btn btn-primary mx-2 mb-3">Create New</a>
        }

        <a asp-action="Index" class="btn btn-secondary mb-3">Table View</a>

        <div class="row">
            @foreach (var item in Model)
            {
                <div class="col-sm-6 col-md-3 col-lg-3 m-3 mx-5 card d-flex justify-content-evenly">
                    <div class="box h-100 text-center card-body">
                        <h5><strong>@item.StudentStatusName</strong></h5>
                        <p>@item.StudentStatusDescription</p>
                        <a asp-action="Edit" asp-route-id="@item.StudentStatusId">Edit</a> |
                        <a asp-action="Delete" asp-route-id="@item.StudentStatusId">Delete</a>
                    </div>
                </div>
            }
        </div>
    </div>
</section>
```

Image Upload:

In this code snippet, the user is given the ability to upload an image directly from their device.



```
#region File Upload - CREATE

if (student.ImageFile != null && student.ImageFile.Length < 4_194_303)
{
    student.PhotoUrl = Guid.NewGuid() + Path.GetExtension(path: student.ImageName);

    string webRootPath = _webHostEnvironment.WebRootPath;
    string fullImagePath = webRootPath + "/assets/img/";
    using var MemoryStream? memoryStream = new MemoryStream();
    await student.ImageFile.CopyToAsync(target: memoryStream);

    using Image img = Image.FromStream(stream: memoryStream);
    ImageUtility.ResizeImage(savePath: fullImagePath, fileName: student.PhotoUrl, image: img, maxImgSize: 300, maxThumbSize: 300);
}
else
{
    student.PhotoUrl = "noimage.png";
}

#endregion
```