

# Red Jaguars Hogwarts Class Scheduling Proposal



## About Us

RedJaguars Inc. is a new startup tackling the web development issues of today's industry. We consist of three team members: Jace Cockrell, Kurtis Caldwell, and Kyle Gough. We are full stack developers with a variety of skillsets including:

.NET Core MVC Framework	Database Management
Full Stack Web Application Design	Unit Testing
C# Fundamentals	Software Development Cycle
Javascript (React)	Agile – Scrum Methodology



## Overview

Hogwarts is an educational complex, offering training for young witches and wizards in the ways of magic to better prepare them to contribute to the wizarding society. Hogwarts administration came to RedJaguars Inc to develop an app for their scheduling staff to help with course organization and enrollment.

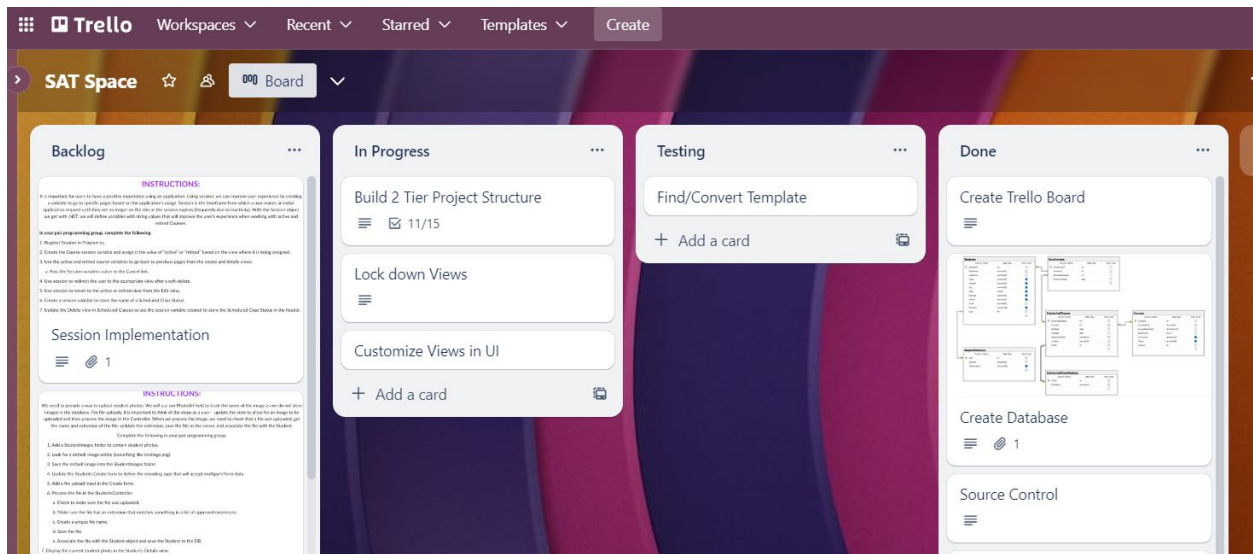
## Requirements

- Tracked progress via Trello board with assigned tasks and cards
- Project history tracking and collaboration via GitHub
- Project documentation via screenshots
- Build relational database for data structure/storage and populate data
- Metadata of classes for data annotation
- Utilize multiple roles for Administrators and Schedulers for authentication
- CRUD functionality across classes, enrollments, students, and statuses for authorized users per Identity Matrix

# Documentation

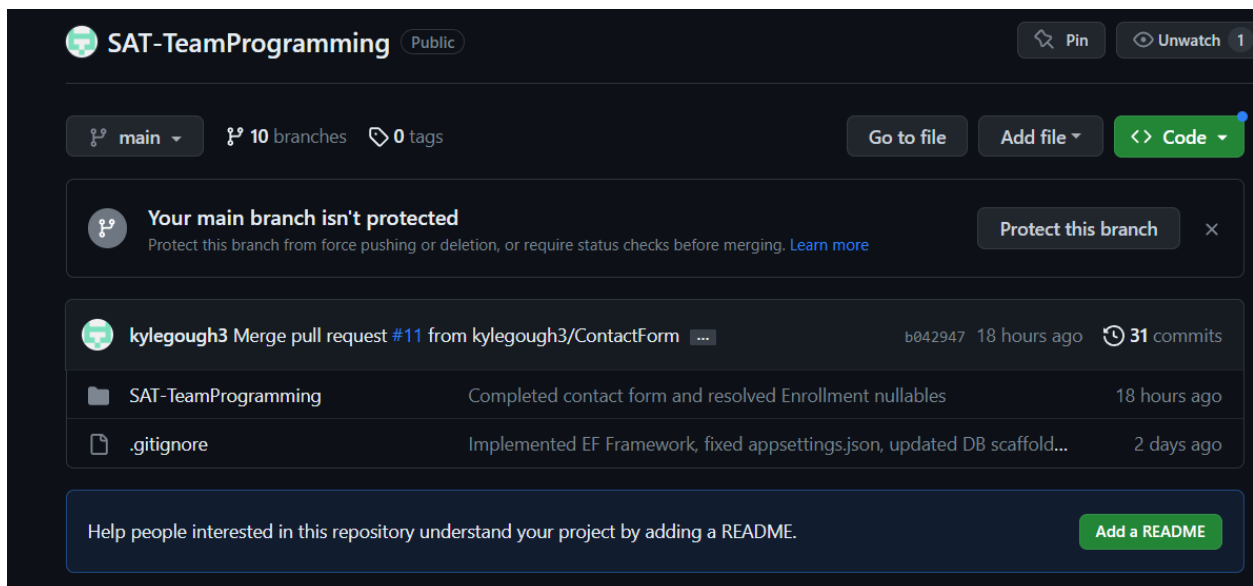
## Trello Board

Trello Boards were utilized to help track allocation of work, requirements, deadlines etc.

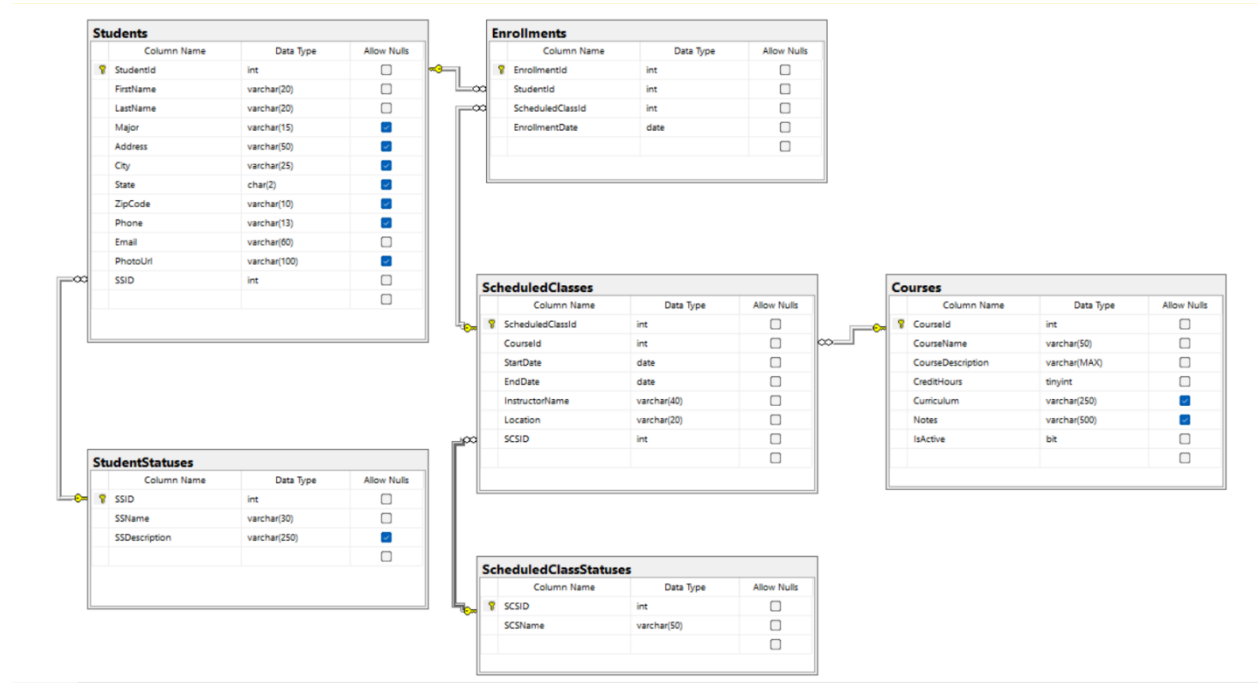


## GitHub – Tracking/Collaboration

GitHub was used to work collaboratively on the codebase, implementing branching techniques in line with current industry standards.



## SQL Database Schema



## Authorization – Identity Matrix

<b>HomeController</b>	Index	Contact
Anonymous	x	x
Scheduling	x	x
Admin	x	x

<b>Courses</b>	Active	Retired	Detail	Create	Edit	Delete
Anonymous						
Scheduling						
Admin	x	x	x	x	x	x

<b>Enrollments</b>	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	x
Admin	x	x	x	x	x

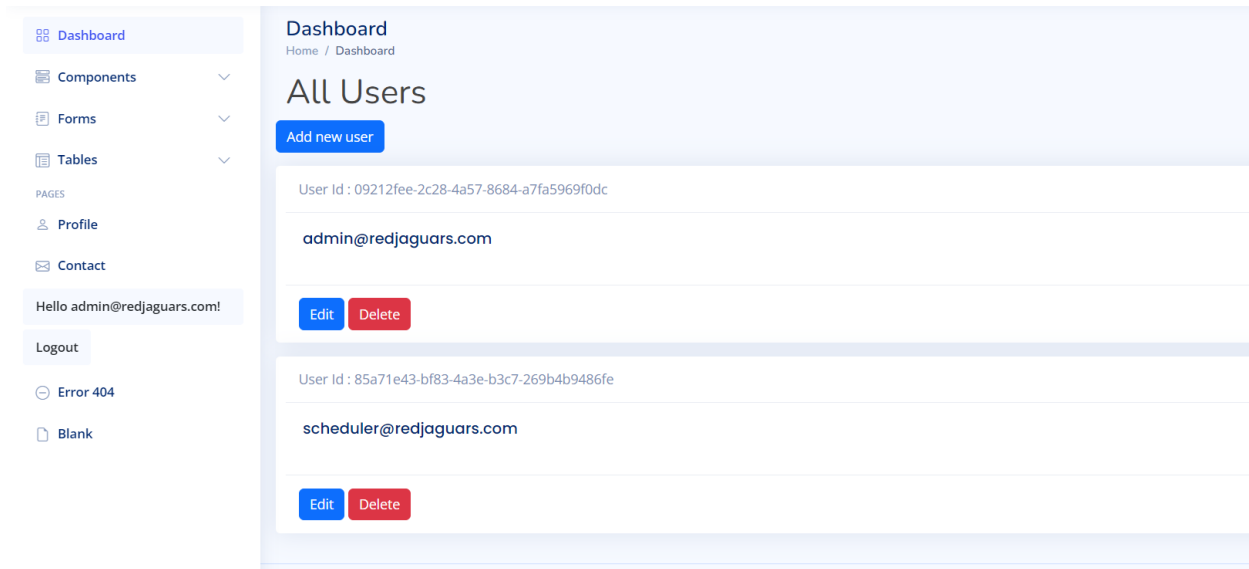
<b>Scheduled Classes</b>	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	
Admin	x	x	x	x	x

<b>Students</b>	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x			
Admin	x	x	x	x	x

<b>Student Statuses</b>	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling					
Admin	x	x	x	x	x

## Identity Management - Code Example

Below shows the admin functionality to manage the users and roles within the site



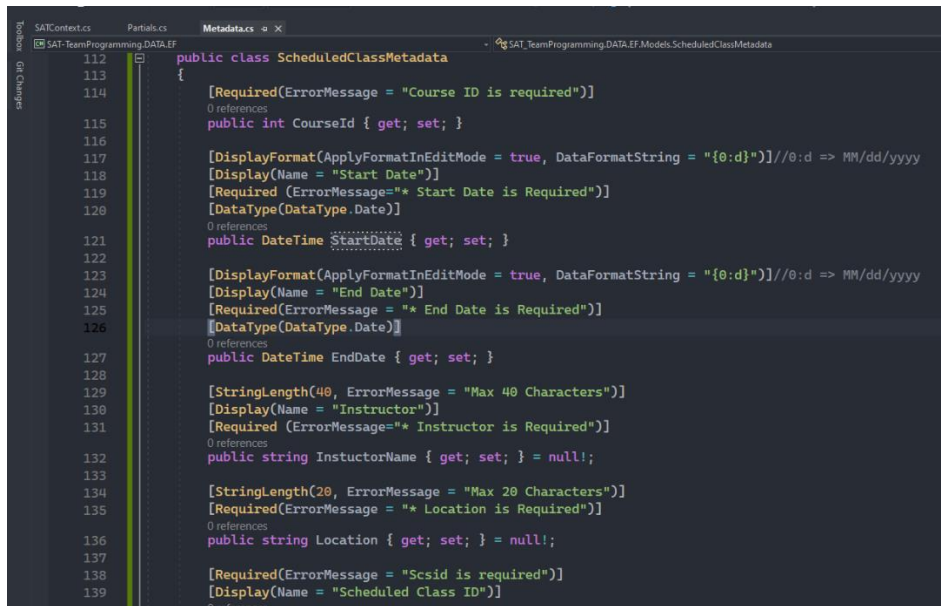
## Authorization – Navigation Code Example

Below is a snip of how the navigation links are limited only to authorized users

```
<i class="bi bi-journal-text"></i><span>Scheduler</span><i class="bi bi-chevron-down ms-auto"></i>
</a>
@if (User.IsInRole("Scheduler") || User.IsInRole("Admin"))
{
    <ul id="forms-nav" class="nav-content collapse " data-bs-parent="#sidebar-nav">
        <li>
            <a href=@Url.Action("Index", "Enrollments")>
                <i class="bi bi-circle"></i><span>Enrollments</span>
            </a>
        </li>
        <li>
            <a href=@Url.Action("Index", "ScheduledClasses")>
                <i class="bi bi-circle"></i><span>Scheduled Classes</span>
            </a>
        </li>
        <li>
            <a href=@Url.Action("Index", "Students")>
                <i class="bi bi-circle"></i><span>Students</span>
            </a>
        </li>
        <li>
            <a asp-action="ListUsers" asp-controller="Administration">
                <i class="bi bi-circle"></i><span>Users</span>
            </a>
        </li>
        <li>
            <a asp-action="ListRoles" asp-controller="Administration">
                <i class="bi bi-circle"></i><span>Roles</span>
            </a>
        </li>
    </ul>
}
```

## Metadata – Data Annotation

These data annotations help protect proper data interaction with the database during create/edit functions



```
112 public class ScheduledClassMetadata
113 {
114     [Required(ErrorMessage = "Course ID is required")]
115     public int CourseId { get; set; }
116
117     [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:d}")]/:/0:d => MM/dd/yyyy
118     [Display(Name = "Start Date")]
119     [Required(ErrorMessage = "* Start Date is Required")]
120     [DataType(DataType.Date)]
121     public DateTime StartDate { get; set; }
122
123     [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:d}")]/:/0:d => MM/dd/yyyy
124     [Display(Name = "End Date")]
125     [Required(ErrorMessage = "* End Date is Required")]
126     [DataType(DataType.Date)]
127     public DateTime EndDate { get; set; }
128
129     [StringLength(40, ErrorMessage = "Max 40 Characters")]
130     [Display(Name = "Instructor")]
131     [Required(ErrorMessage = "* Instructor is Required")]
132     public string InstructorName { get; set; } = null;
133
134     [StringLength(20, ErrorMessage = "Max 20 Characters")]
135     [Required(ErrorMessage = "* Location is Required")]
136     public string Location { get; set; } = null;
137
138     [Required(ErrorMessage = "Scsid is required")]
139     [Display(Name = "Scheduled Class ID")]
```

## Partial Class – Custom Properties

These custom properties allow for additional data to be rendered during the CRUD functions.



```

[ModelMetadataType(typeof(StudentMetadata))]
24 references
public partial class Student
{
    //Added to use for Full Name display
    2 references
    public string FullName { get { return $"{FirstName} {LastName}"; } }
}

[ModelMetadataType(typeof(StudentStatusMetadata))]
22 references
public partial class StudentStatus
{
}

[ModelMetadataType(typeof(EnrollmentMetadata))]
21 references
public partial class Enrollment
{
}

[ModelMetadataType(typeof(ScheduledClassMetadata))]
26 references
public partial class ScheduledClass
{
    2 references
    public string? SchClassInfo
    {
        get
        { if(Course == null)
          {
              return null;
          } else
          {
              return string.Format($"{Course.CourseName} - {StartDate:d}");
          }
        }
    }
}
```

## Contact Page

Full functionality contact page with Confirmation page upon send

Dashboard

Admin

Scheduler

PAGES

Courses

Contact

Hello admin@redjaguars.com!

Logout

Dashboard

Home / Dashboard

Contact

Name

Email

Subject

Message

Send It

Back to Home

Dashboard

Home / Dashboard

Message Sent!

Thank you for contacting us! We will respond to your message using the provided email address, **bro.bro@bro.org**, as soon as we are able.

You sent the following message on 7/21/2023 at 10:56 AM

Bruh