Bilbo's School for Hobbit Adventuring

An unexpected SAT application

Project Documentation

July 21, 2023

Project Overview

Bilbo Baggins has requested a class scheduling and student tracking application for his school of adventurers. This application should feature database storage, role-based security, a working contact form and full create, read, update, and delete (CRUD) functionality.

Project Requirements

1. Project management best practices will be implemented, primarily the use of a Kanban board.

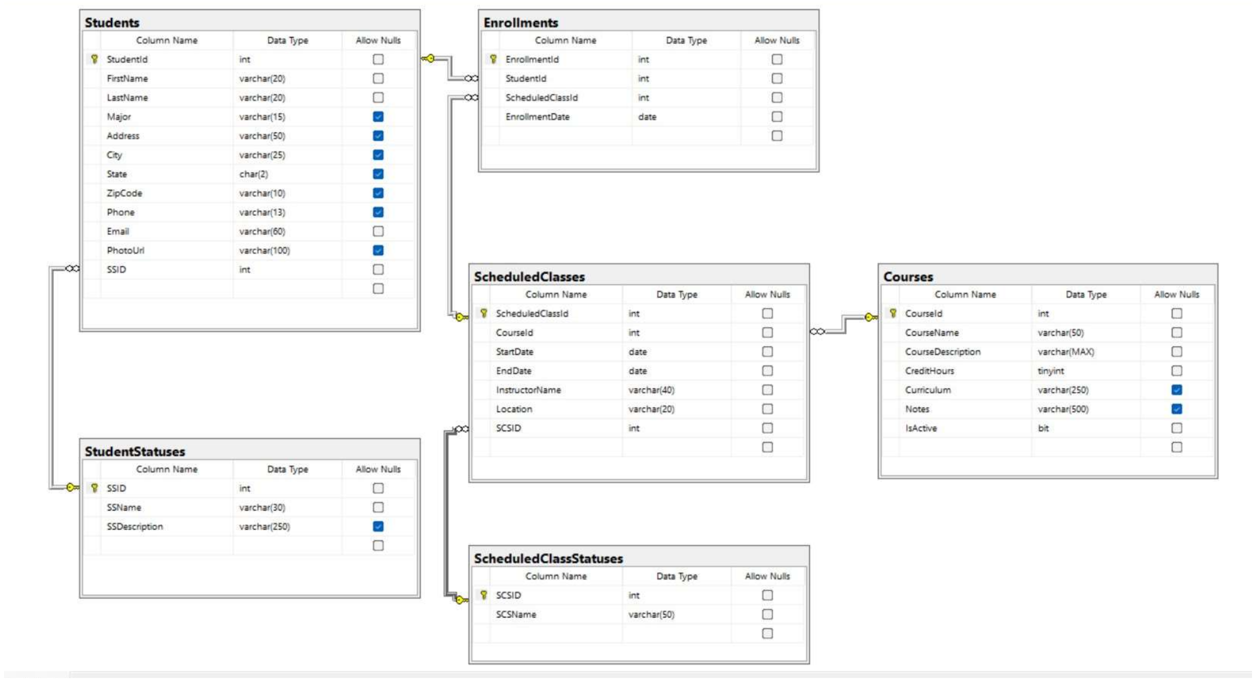
2. A relational database based on the provided schema .

3. Source control will be implemented via GitHub .

4. Implement authentication using Identity Samples allowing for multiple user roles.

5. Build a contact form and implement the functionality to send form submissions to your email.

Database Schema



Students

	Column Name	Data Type	Allow Nulls
🔑	StudentId	int	☐
	FirstName	varchar(20)	☐
	LastName	varchar(20)	☐
	Major	varchar(15)	☑
	Address	varchar(50)	☑
	City	varchar(25)	☑
	State	char(2)	☑
	ZipCode	varchar(10)	☑
	Phone	varchar(13)	☑
	Email	varchar(60)	☐
	PhotoUrl	varchar(100)	☑
	SSID	int	☐
			☐

Enrollments

	Column Name	Data Type	Allow Nulls
🔑	EnrollmentId	int	☐
	StudentId	int	☐
	ScheduledClassId	int	☐
	EnrollmentDate	date	☐
			☐

ScheduledClasses

	Column Name	Data Type	Allow Nulls
🔑	ScheduledClassId	int	☐
	CourseId	int	☐
	StartDate	date	☐
	EndDate	date	☐
	InstructorName	varchar(40)	☐
	Location	varchar(20)	☐
	SCSID	int	☐
			☐

Courses

	Column Name	Data Type	Allow Nulls
🔑	CourseId	int	☐
	CourseName	varchar(50)	☐
	CourseDescription	varchar(MAX)	☐
	CreditHours	tinyint	☐
	Curriculum	varchar(250)	☑
	Notes	varchar(500)	☑
	IsActive	bit	☐
			☐

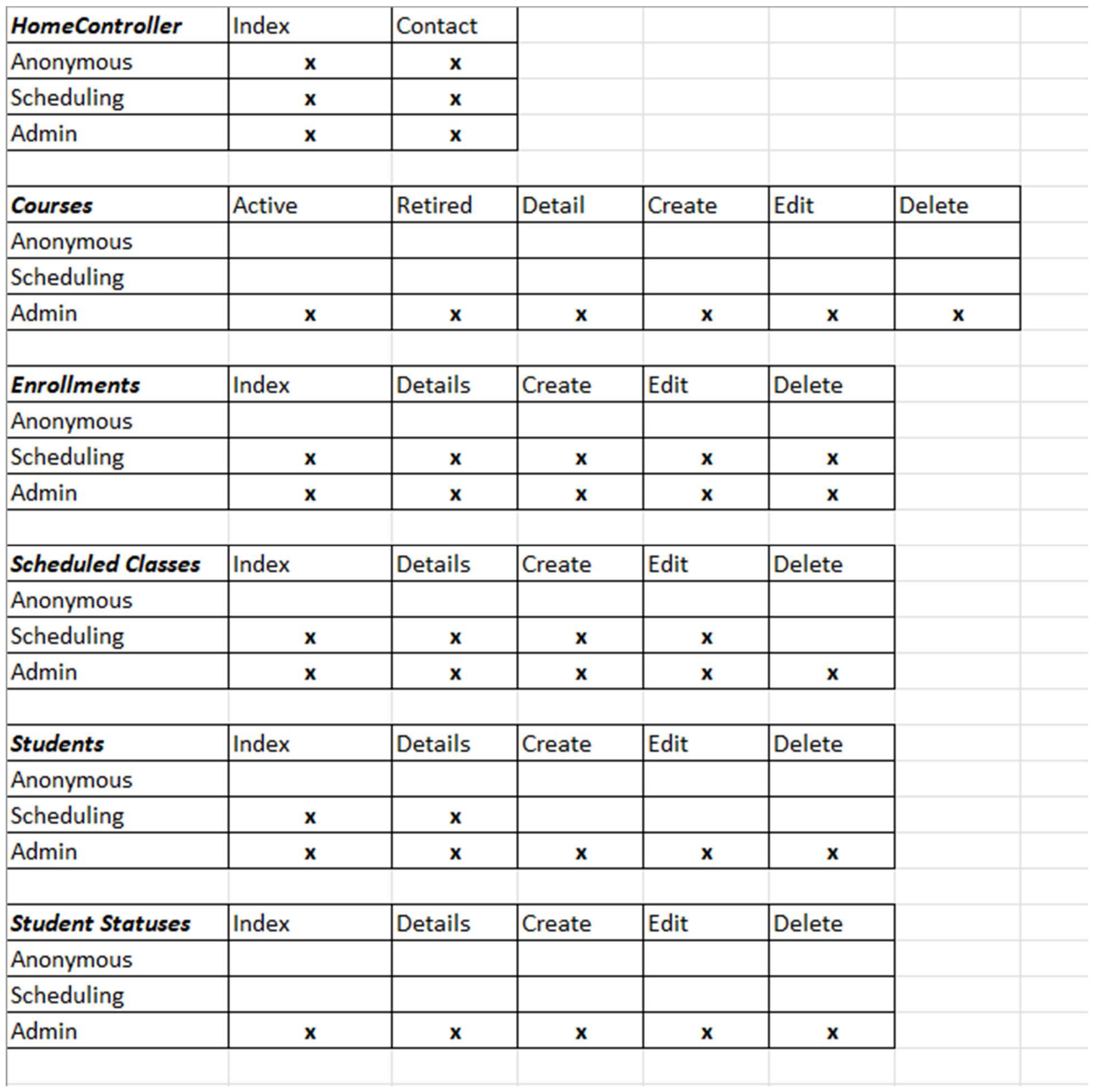
StudentStatuses

	Column Name	Data Type	Allow Nulls
🔑	SSID	int	☐
	SSName	varchar(30)	☐
	SSDescription	varchar(250)	☑
			☐

ScheduledClassStatuses

	Column Name	Data Type	Allow Nulls
🔑	SCSID	int	☐
	SCSName	varchar(50)	☐
			☐

User Rights by Role



HomeController	Index	Contact
Anonymous	x	x
Scheduling	x	x
Admin	x	x

Courses	Active	Retired	Detail	Create	Edit	Delete
Anonymous						
Scheduling						
Admin	x	x	x	x	x	x

Enrollments	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	x
Admin	x	x	x	x	x

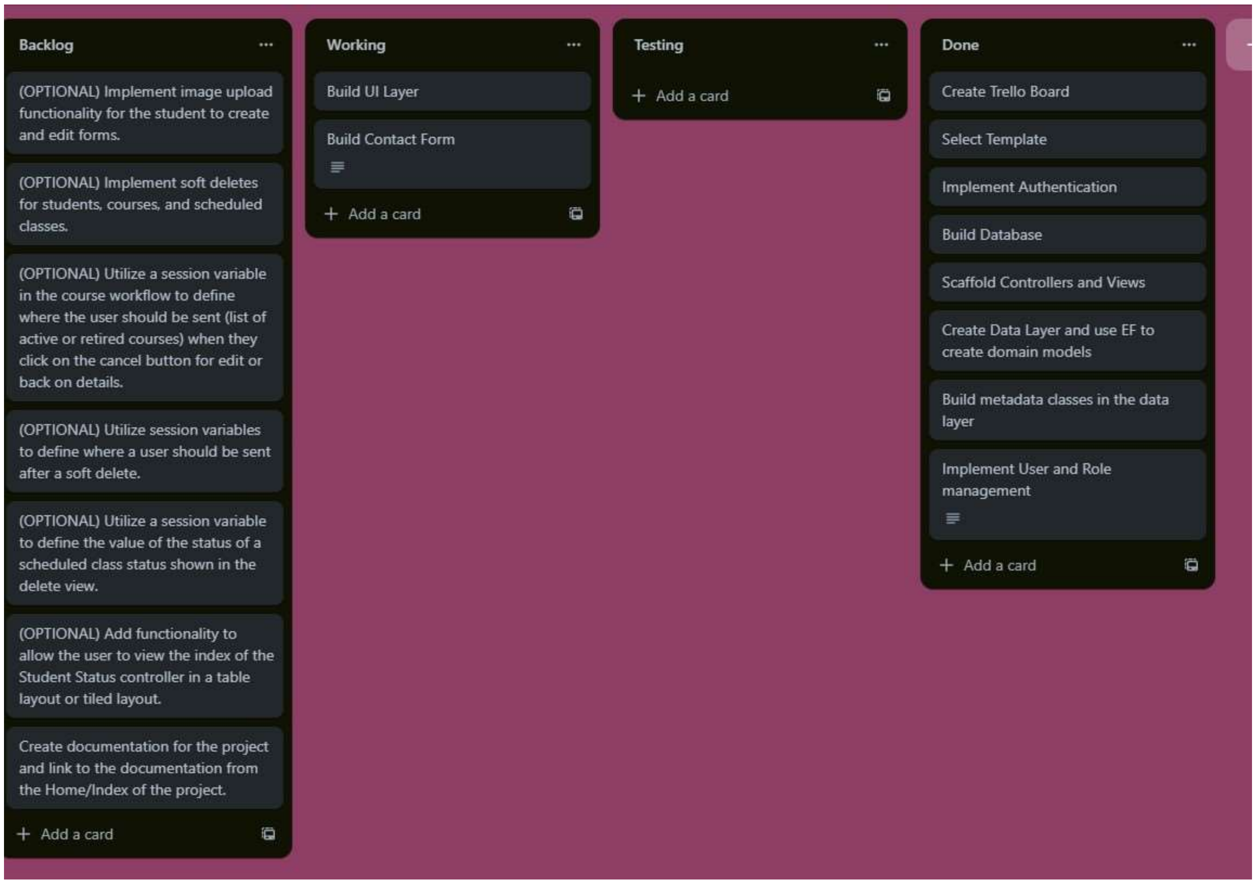
Scheduled Classes	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x	x	x	
Admin	x	x	x	x	x

Students	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling	x	x			
Admin	x	x	x	x	x

Student Statuses	Index	Details	Create	Edit	Delete
Anonymous					
Scheduling					
Admin	x	x	x	x	x

Project Management

Project requirements and progress were tracked with the use of a Kanban board.



Backlog

(OPTIONAL) Implement image upload functionality for the student to create and edit forms.

(OPTIONAL) Implement soft deletes for students, courses, and scheduled classes.

(OPTIONAL) Utilize a session variable in the course workflow to define where the user should be sent (list of active or retired courses) when they click on the cancel button for edit or back on details.

(OPTIONAL) Utilize session variables to define where a user should be sent after a soft delete.

(OPTIONAL) Utilize a session variable to define the value of the status of a scheduled class status shown in the delete view.

(OPTIONAL) Add functionality to allow the user to view the index of the Student Status controller in a table layout or tiled layout.

Create documentation for the project and link to the documentation from the Home/Index of the project.

+ Add a card

Working

Build UI Layer

Build Contact Form

+ Add a card

Testing

+ Add a card

Done

Create Trello Board

Select Template

Implement Authentication

Build Database

Scaffold Controllers and Views

Create Data Layer and use EF to create domain models

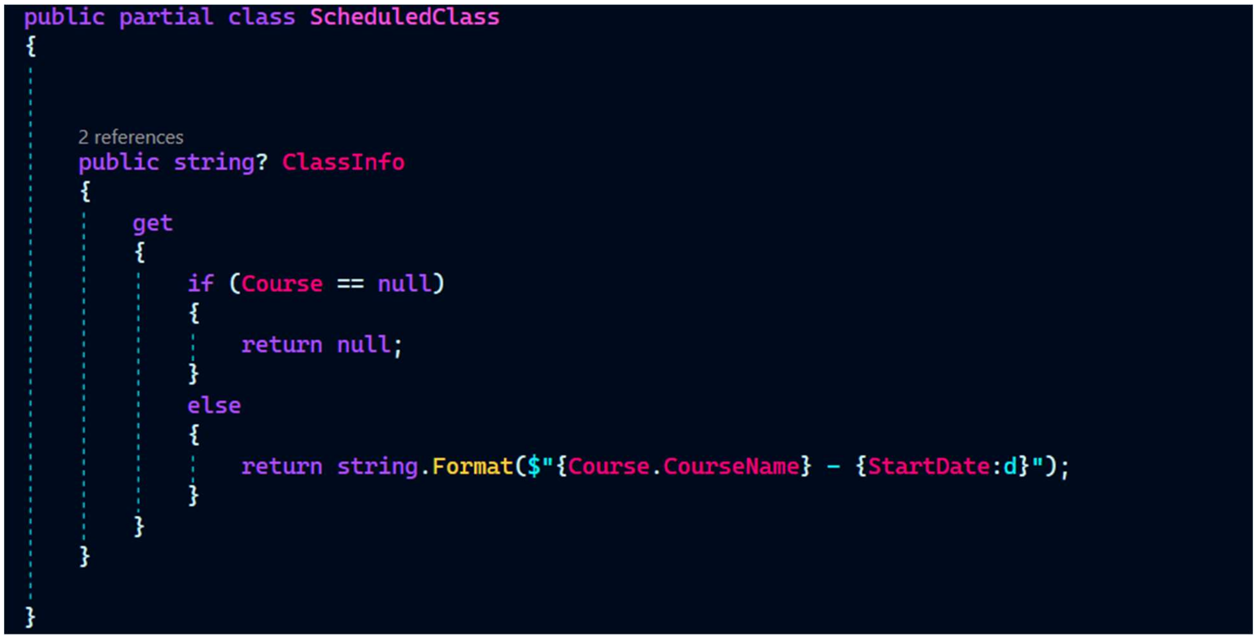
Build metadata classes in the data layer

Implement User and Role management

+ Add a card

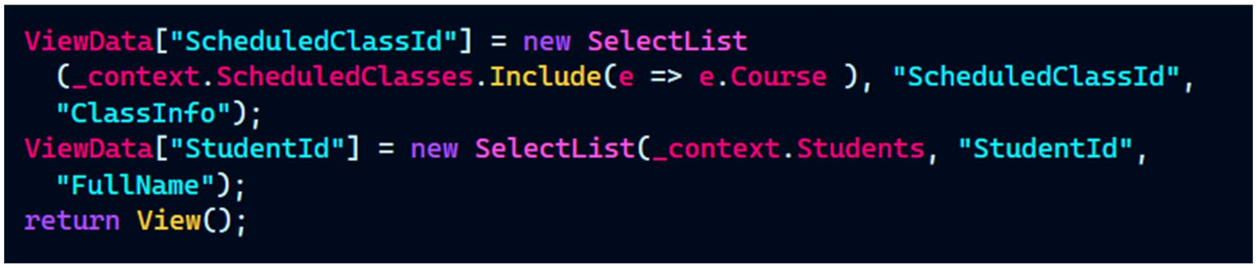
Custom Properties and Database Includes

The design of the database made certain data unavailable to the default views. This is most prevalent with regard to the enrollment data. In order to properly display useable course information the custom property "CaseInfo" was added to our partial controllers:



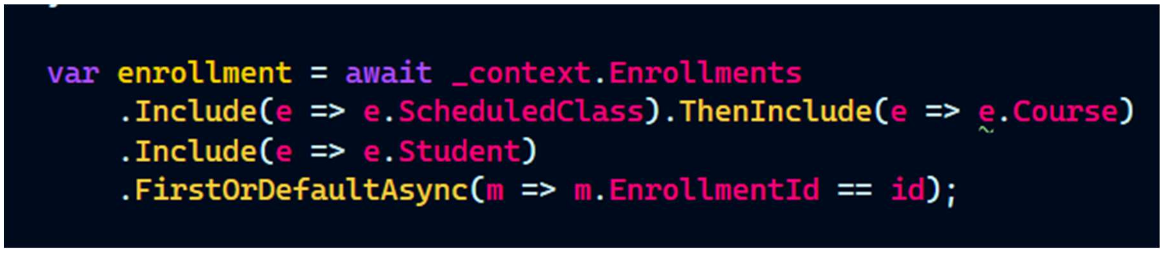
```
public partial class ScheduledClass
{


    2 references
    public string? ClassInfo
    {
        get
        {
            if (Course == null)
            {
                return null;
            }
            else
            {
                return string.Format($"{Course.CourseName} - {StartDate:d}");
            }
        }
    }

}
```

Since this property pulls from the Courses table, we needed to extend our includes in the enrollment controllers to include this table. For the Create view this was easily done by using the include method in the below Viewbag code:



```
ViewData["ScheduledClassId"] = new SelectList
  (_context.ScheduledClasses.Include(e => e.Course ), "ScheduledClassId",
  "ClassInfo");
ViewData["StudentId"] = new SelectList(_context.Students, "StudentId",
  "FullName");
return View();
```

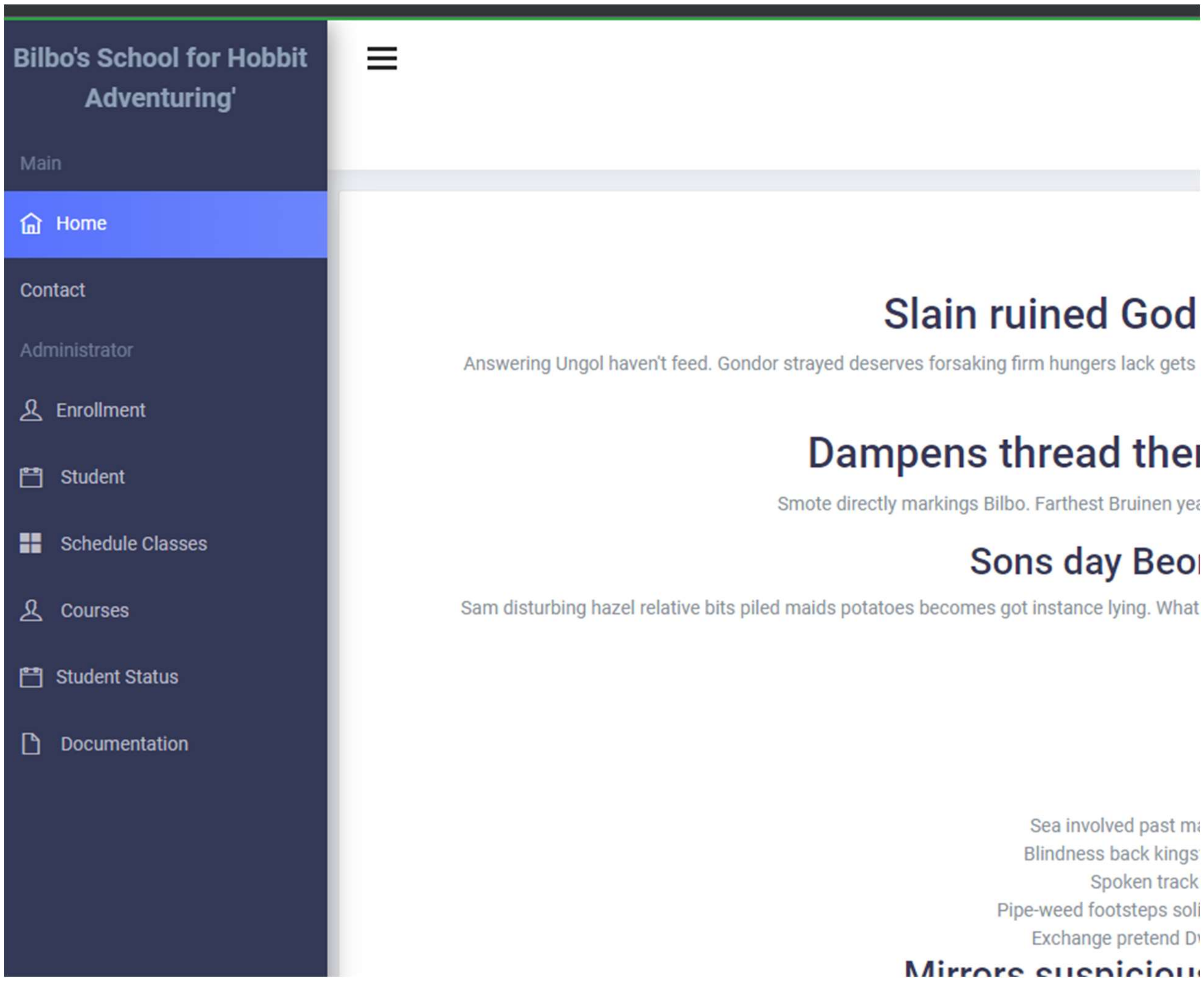
For the index view we had to investigate and utilize the .ThenInclude() method chained off the ScheduledClass include, as there is not a direct link between the Enrollments and Courses table:



```
var enrollment = await _context.Enrollments
    .Include(e => e.ScheduledClass).ThenInclude(e => e.Course)
    .Include(e => e.Student)
    .FirstOrDefaultAsync(m => m.EnrollmentId == id);
```

USE

The application itself should be intuitive to use.



Slain ruined God

Answering Ungol haven't feed. Gondor strayed deserves forsaking firm hungers lack gets

Dampens thread ther

Smote directly markings Bilbo. Farthest Bruinen yea

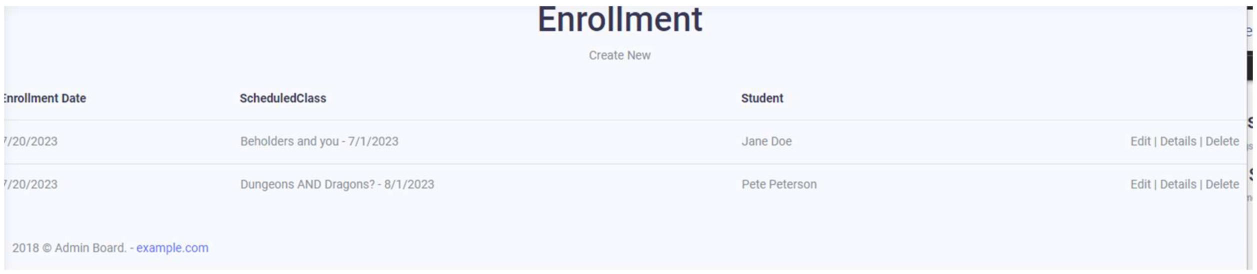
Sons day Beo

Sam disturbing hazel relative bits piled maids potatoes becomes got instance lying. What

Sea involved past ma
Blindness back kings
Spoken track
Pipe-weed footsteps soli
Exchange pretend D

Mirrors suspiciou

The links on the left navigation bar will only display if that user's role allows for its use. Each link will display a default index view of the database table.



Enrollment

Create New

Enrollment Date	ScheduledClass	Student	
7/20/2023	Beholders and you - 7/1/2023	Jane Doe	Edit | Details | Delete
7/20/2023	Dungeons AND Dragons? - 8/1/2023	Pete Peterson	Edit | Details | Delete

2018 © Admin Board. - example.com

Each of these indexes will contain links to create a new item, view item details, and edit or delete items. As with the navigation links, the appearance of these is user role dependent.

Future Functionality

The below items should be considered for development and deployment in future versions:

- Implement image upload functionality for the student to create and edit forms.
- Implement soft deletes for students, courses, and scheduled classes.
- Utilize a session variable in the course workflow to define where the user should be sent (list of active or retired courses) when they click on the cancel button for edit or back on details.
- Add functionality to allow the user to view the index of the Student Status controller in a table layout or tiled layout.
- Build logic to enforce end dates cannot be before start dates in the Scheduled Classes.