Table of contents

[Report 1](#_Toc150454129)

[Accessing the Website 1](#_Toc150454130)

[Website Layout 2](#_Toc150454131)

[Home Page 2](#_Toc150454132)

[Login Page 3](#_Toc150454133)

[Management Page 4](#_Toc150454134)

[Utilizing Website Features 5](#_Toc150454135)

[Login and Registration 5](#_Toc150454136)

[Management 5](#_Toc150454137)

[Technical Demonstrations 5](#_Toc150454138)

[Models 5](#_Toc150454139)

[User and UserLogin 6](#_Toc150454140)

[Student 7](#_Toc150454141)

[StudentIndexViewModel 8](#_Toc150454142)

[CSS 8](#_Toc150454143)

[JavaScript 10](#_Toc150454144)

[Feature Implementation 10](#_Toc150454145)

[Login and Registration 11](#_Toc150454146)

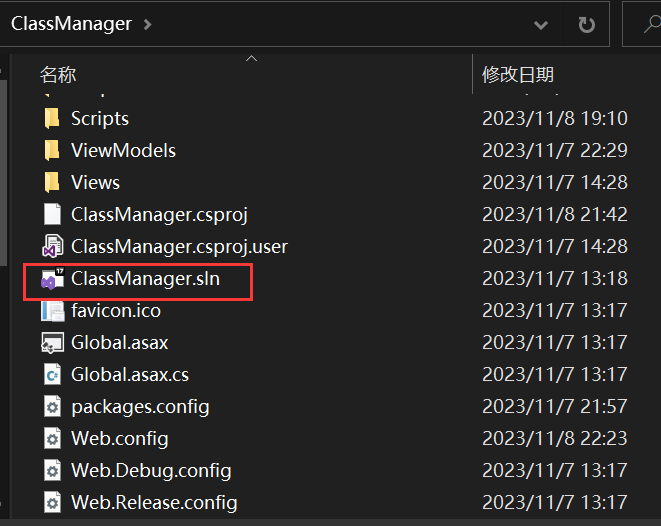
[Management 13](#_Toc150454147)

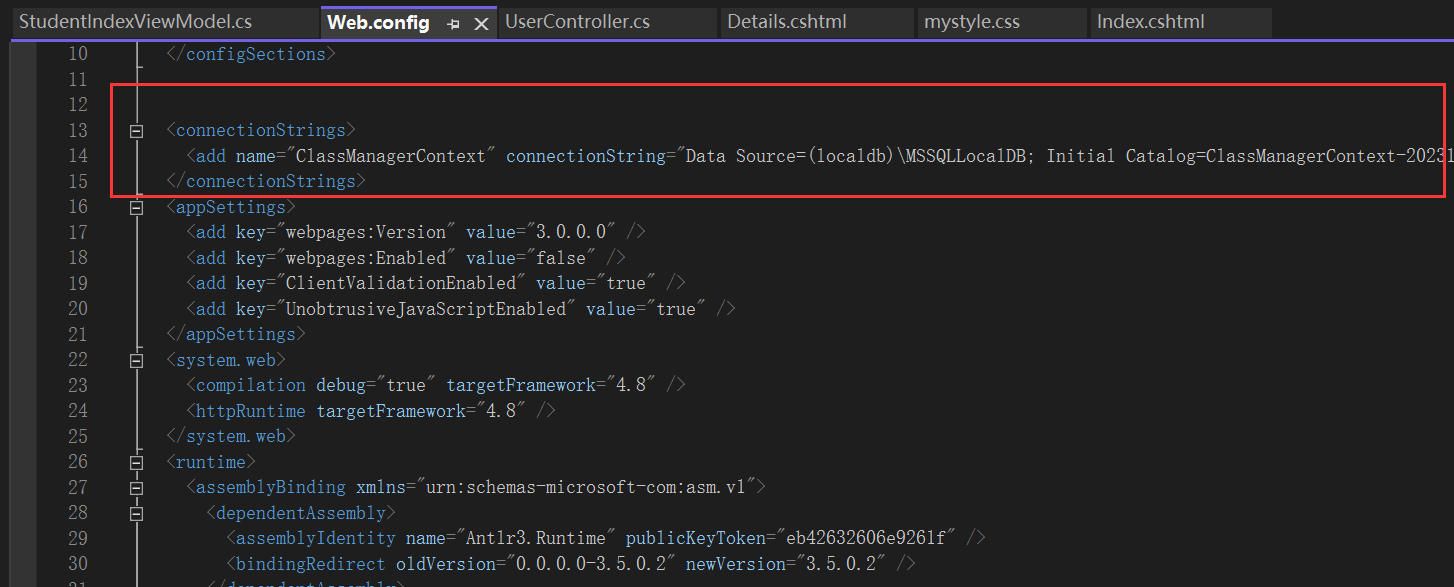
Report

# Accessing the Website

To run the homework assignment, unzip the folder and open the ClassManager.sln file in Visual Studio. Before running, you need to configure the database connection string in the Web.config file. You can use the test data seeds provided in the file to populate the database with test data.

The website is designed to help teachers manage student information. Teachers can register and log in to their account to manage their students' information. They can add, delete, or modify student information, and they can also filter and search student information to quickly find the information they need.



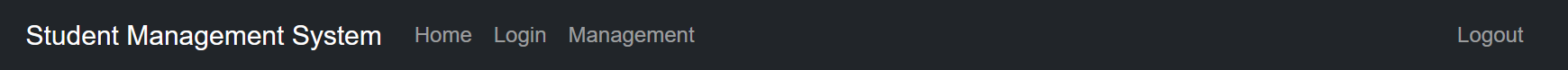


# Website Layout

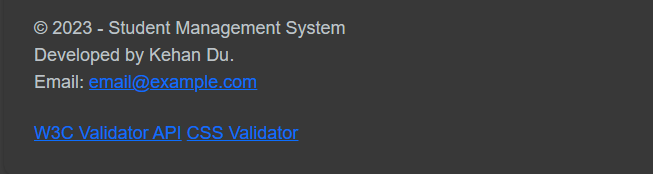
The website's layout is mainly composed of three parts: **the home page**, **the login and registration interface**, and **the student information management interface**.

All web pages are composed of a navigation bar, a body, and a footer. All footers contain links to HTML5 and CSS3 validators as well as an email link.

Navigation bar:

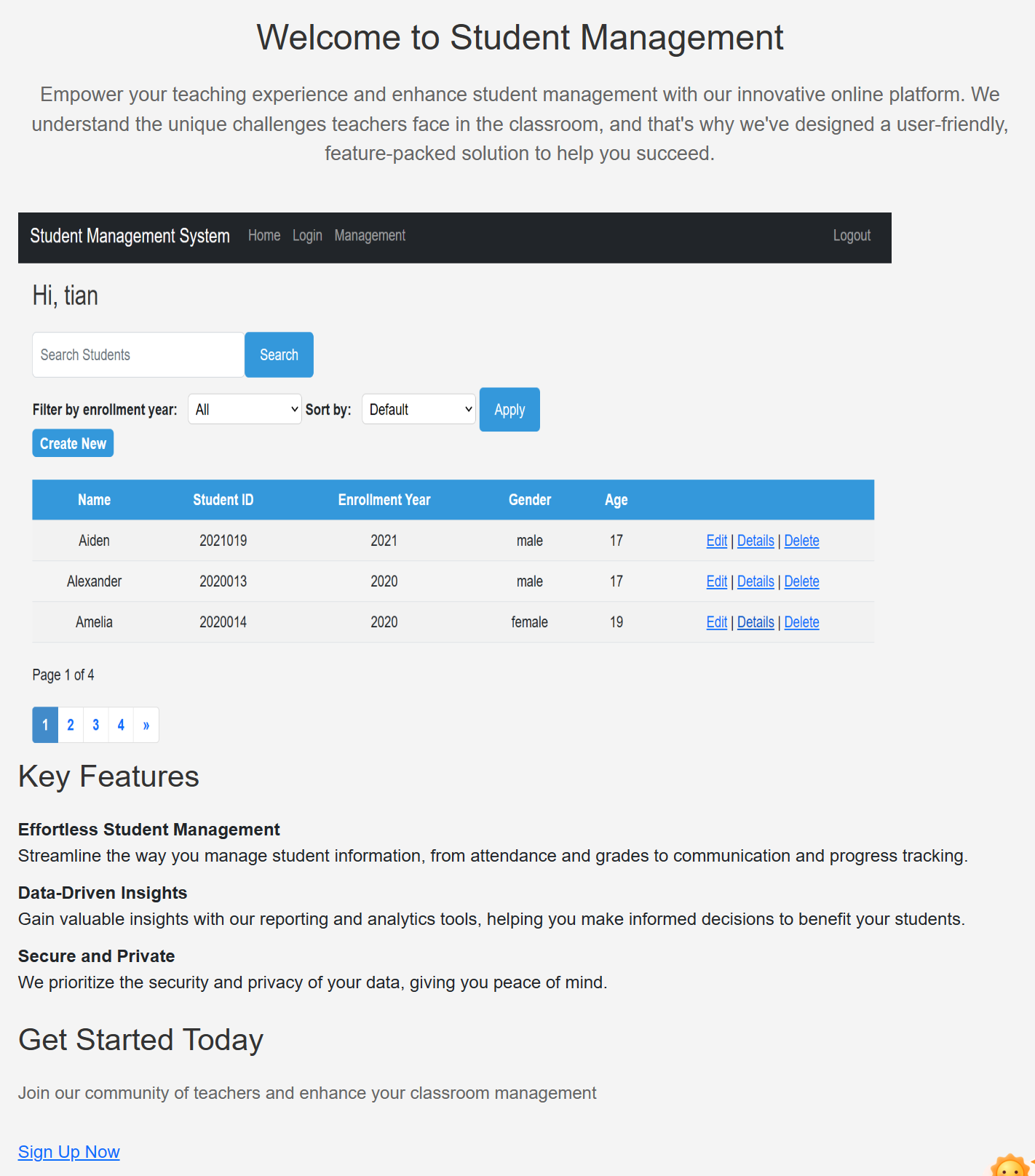


Footer:



## Home Page

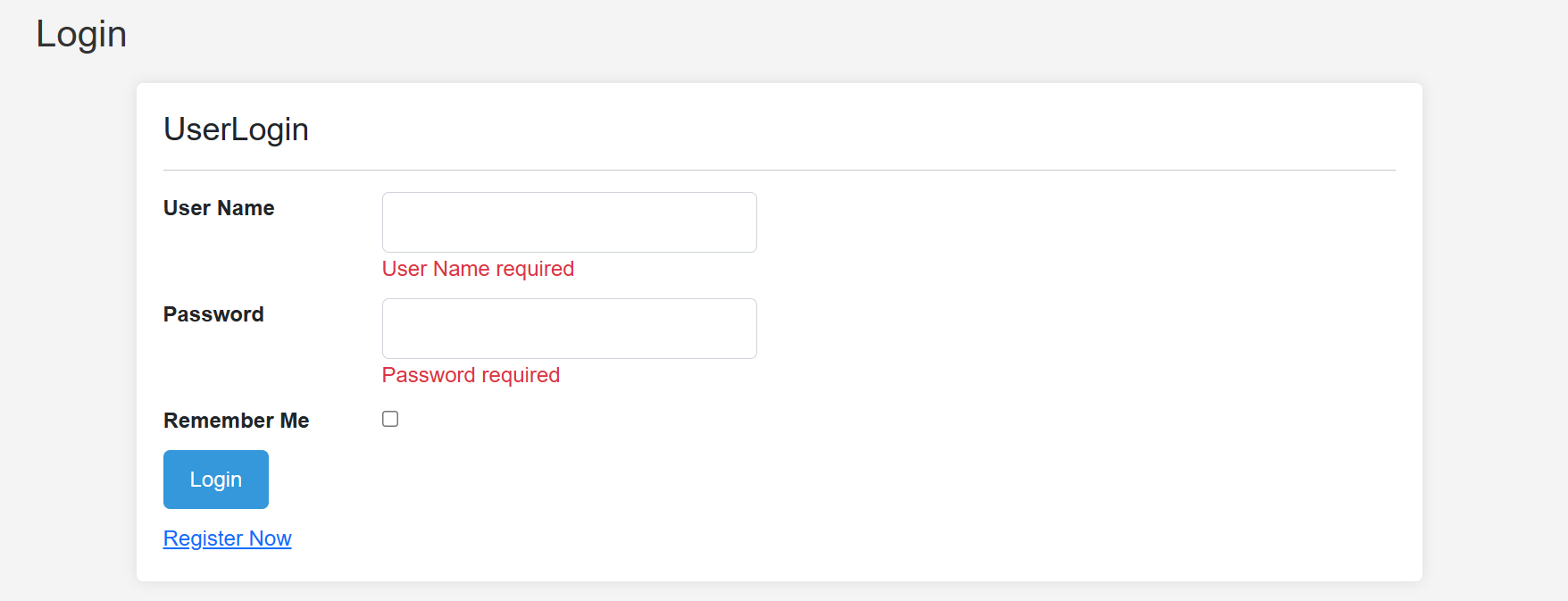
The home page is used to introduce the main functions of a website and attract users. The home page introduces the features of the website and guides users on how to use it.

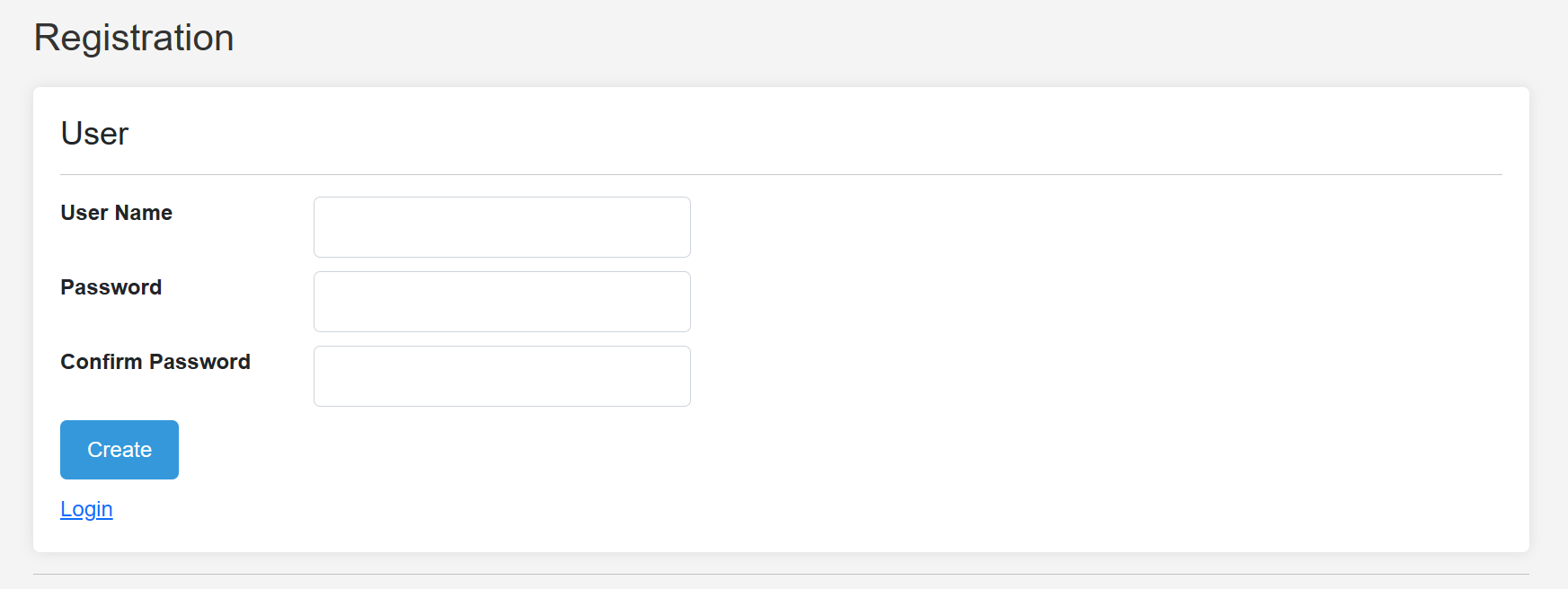


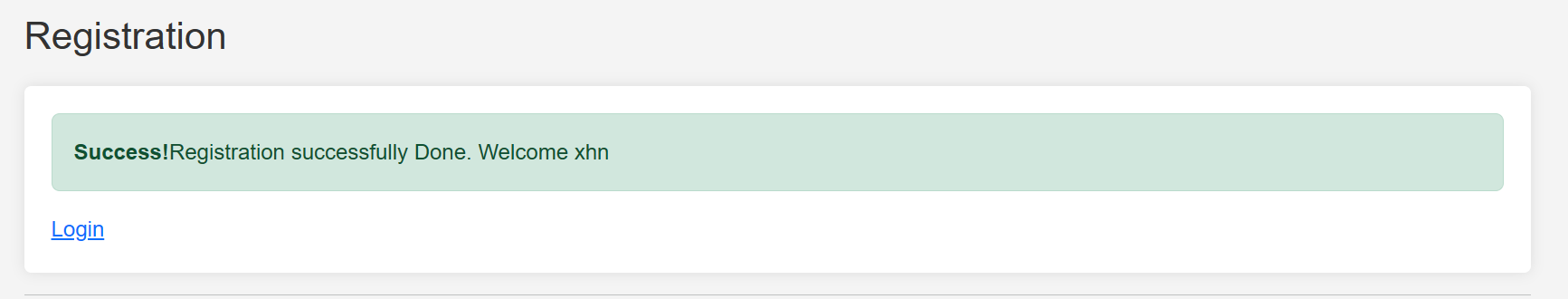
## Login Page

There are two ways to enter the login and registration interface: click the "Sign Up Now" button at the bottom of the homepage or the "Login" link in the navigation bar.

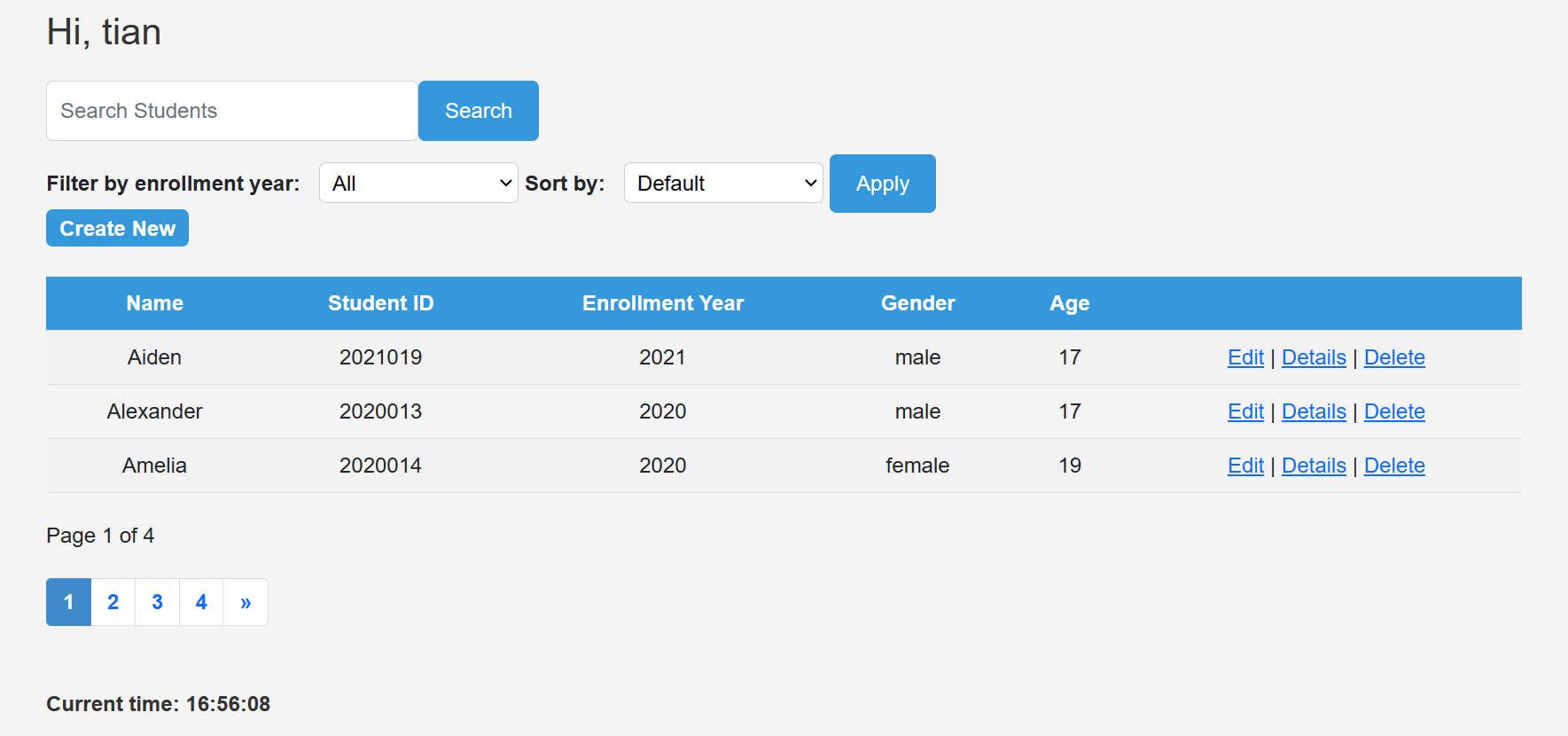
On the login page, you can enter your username and password to log in. If you don't have an account, you can click the "Register Now" button below to enter the registration page and create an account. If there are any errors, such as entering the login page after logging in or trying to register an existing account, the website will give a prompt. After logging in successfully, a "Logout" button will appear in the navigation bar on the far right, which can be used to log out of the current account.







## Management Page



In the management interface, teachers (users) can manage their students' information. Users can easily add, modify, delete, or view a student's information. This interface also includes search, filter, sort, or pagination functions. The interface also includes a welcome message and a clock that displays the current time at the bottom of the page.

# Utilizing Website Features

## Login and Registration

Logging in and registering are straightforward, just like on other websites. To log in, go to the login page from the bottom or top of the main page. To register, simply enter your username, password, and confirm your password. Once you have an account, you can log in.

## Management

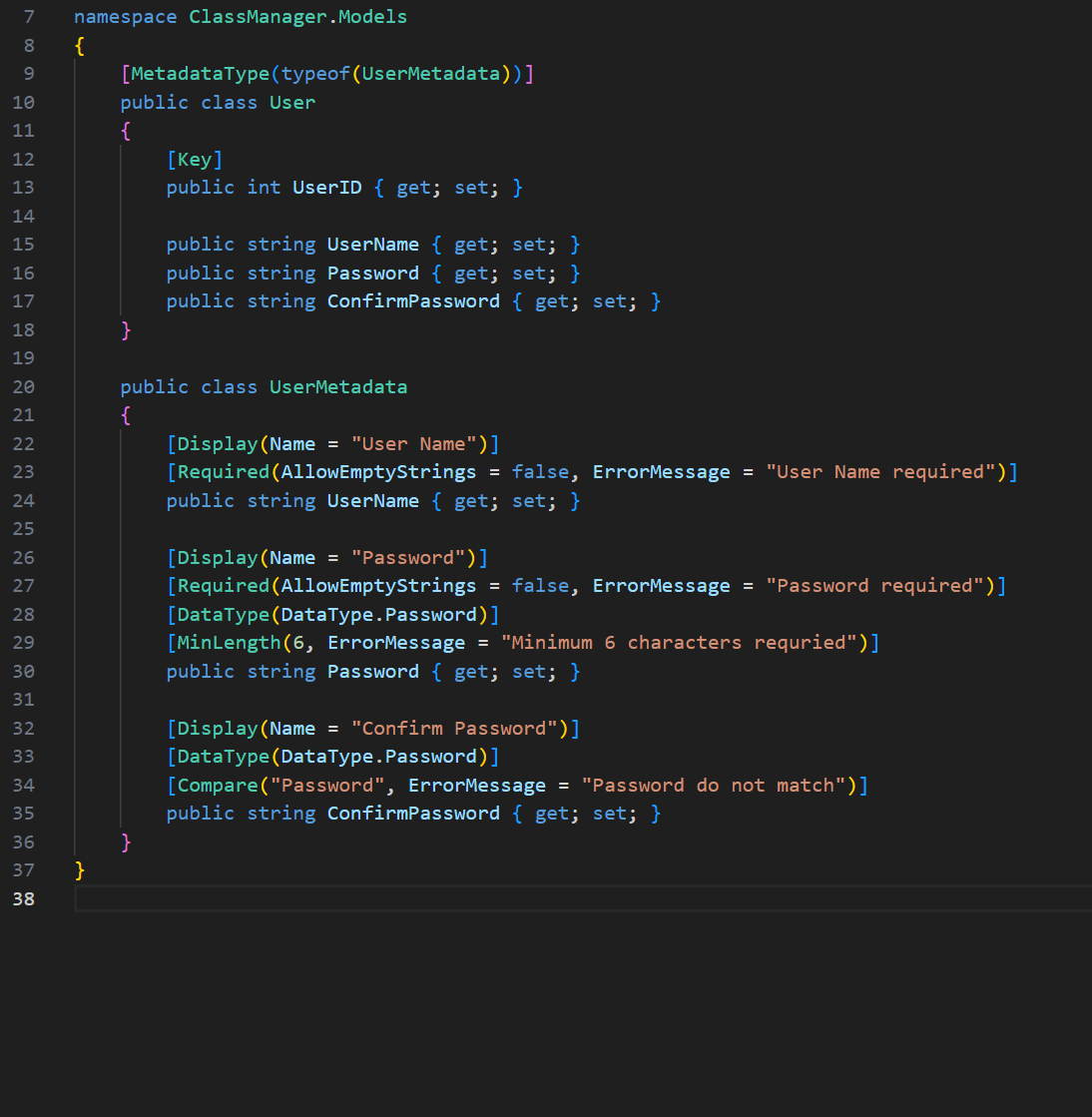
The management interface is intuitive and easy to use. You can access all the features you need by simply clicking the corresponding buttons. For example, you can search for students by name or student ID, or you can filter students by year of enrollment. You can also sort students by student ID.

# Technical Demonstrations

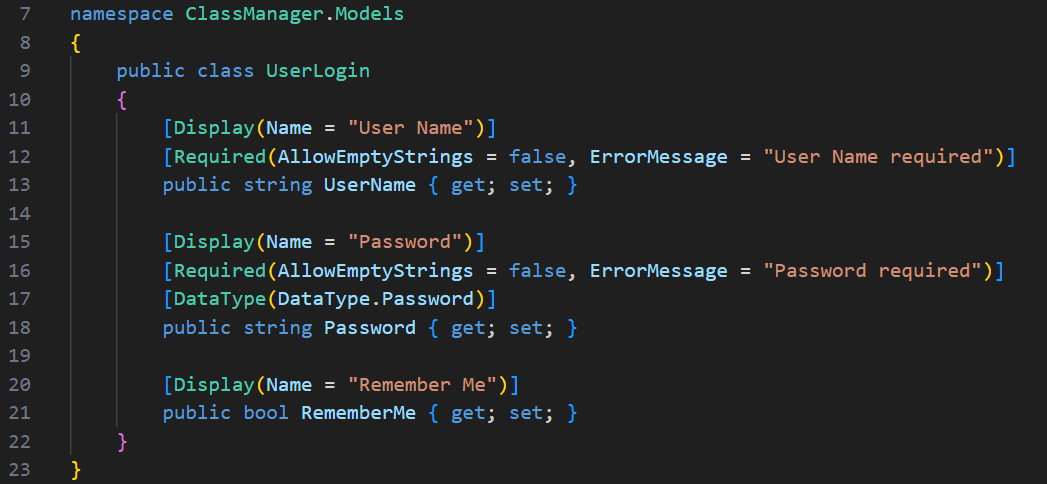
## Models

The project has three main models, Student, User, and UserLogin, and one view model, StudentIndexViewModel.

### User and UserLogin

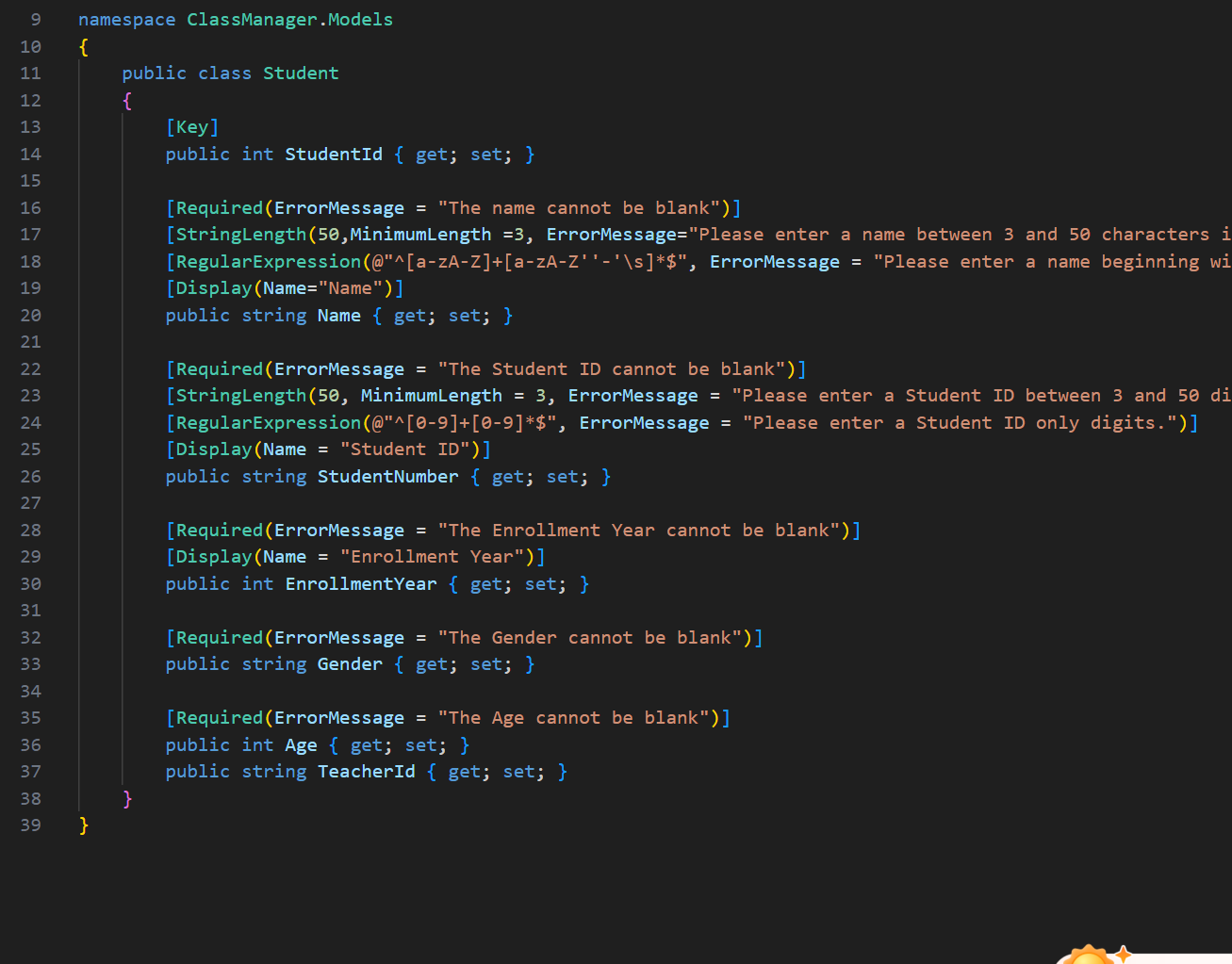


The User model stores user information, including the user ID, name, password, and confirmed password. It also enforces some input validation rules, such as not allowing empty strings and requiring passwords to be at least 6 characters long.



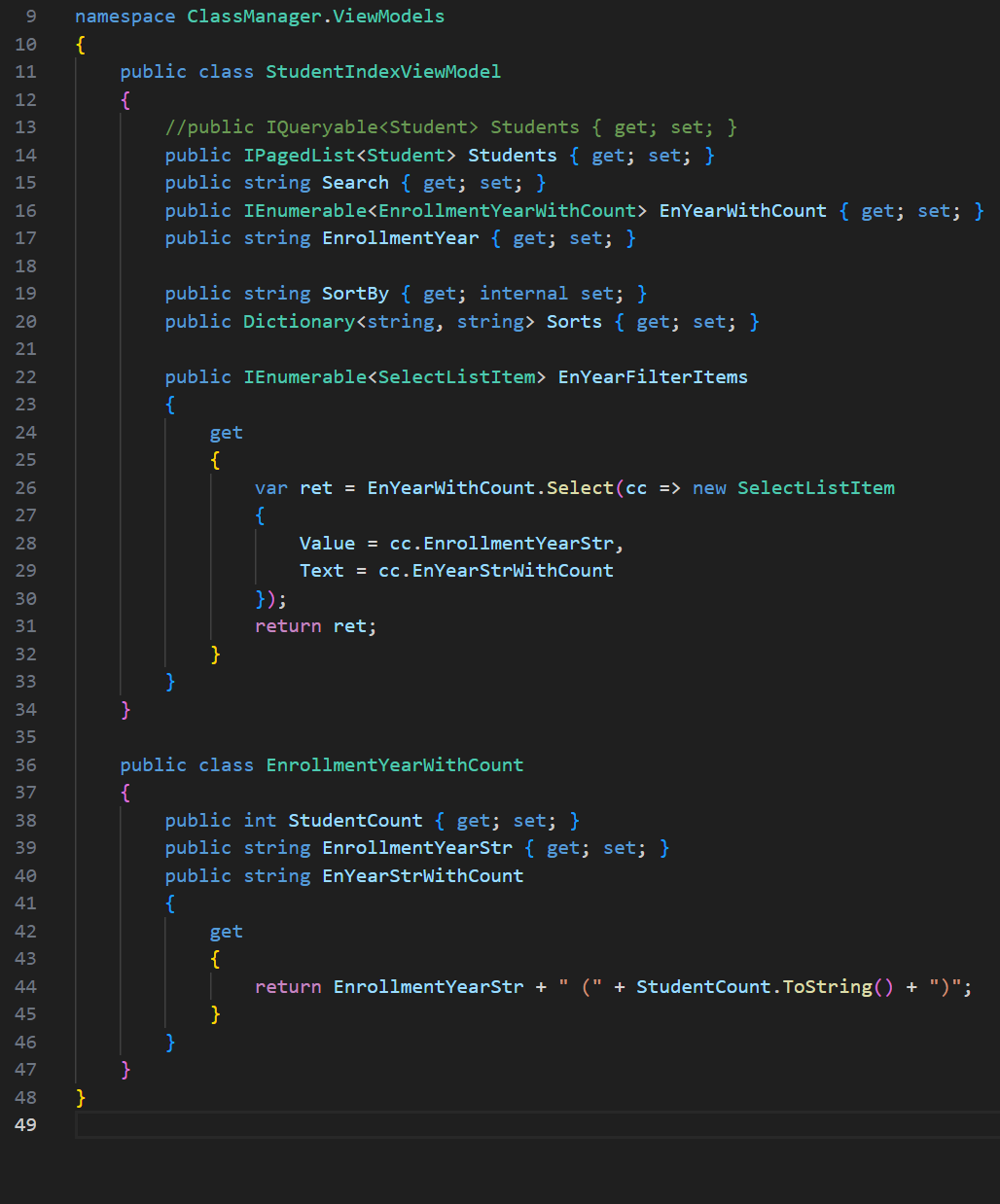
The UserLogin model stores user login information temporarily. It includes the user's name, password, and a flag indicating whether the browser should remember the user's login information.

### Student



The Student model contains basic information about students, such as their name, student ID, gender, enrollment year, age, and the teacher ID who created the student information.

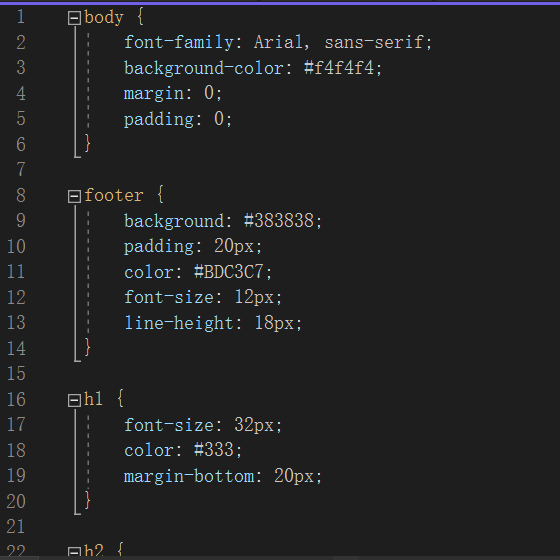
### StudentIndexViewModel



StudentIndexViewModel is a model that helps to display content. It stores the search information, filter information, and sorting order.

## CSS

ClassManager\Content\mystyle.css



Overall Styles:

Clean and modern look with Arial font family.

Light gray background (#f4f4f4) for easy viewing.

No unnecessary whitespace with margins and padding set to 0.

Footer Styles:

Dark gray background (#383838) for contrast.

Padding of 20px for spacing.

Easy-to-read light gray text (#BDC3C7).

Small font size (12px) for compact display.

Line height of 18px for readability.

Heading Styles:

H1: Large and prominent with font size of 32px, dark gray color (#333), and 20px margin-bottom.

H2: Slightly smaller but still noticeable with font size of 28px, dark gray color (#333), and 20px margin-bottom.

Paragraph Styles:

No unnecessary whitespace with margin and padding set to 0.

Main Content Styles:

Maximum width of 960px to prevent excessive width on large screens.

Centered on the page with margin of 0 auto.

Spacing with padding of 20px.

Row Styles:

Centered content with text-align of center.

Lead Paragraph Styles:

Slightly larger font size (18px) compared to regular paragraph text.

Gray color (#666) for subtle contrast.

40px margin-bottom for spacing.

Form Styles:

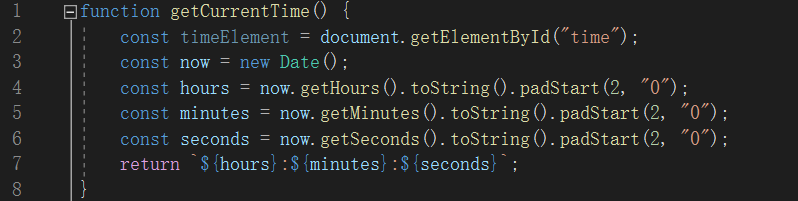
Horizontal arrangement of form controls using flex display property for form groups (.form-group).

10px margin-bottom for spacing between form groups.

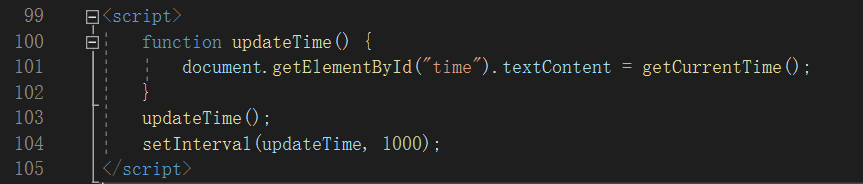
## JavaScript

The JavaScript in this assignment is relatively light-weight. The two main scripts are used to display the current time in real time and to provide error messages.

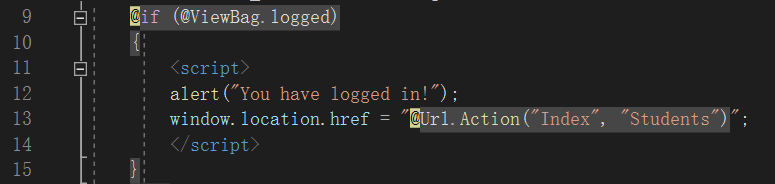
ClassManager\Scripts\getCurrentTime.js



ClassManager\Views\Students\Index.cshtml



ClassManager\Views\User\Login.cshtml



## Feature Implementation

The main functions of this website are: login and registration, student information management (including adding, modifying, viewing, and deleting), search, filter, sort, and pagination.

### Login and Registration

The implementation of the registration and login features is mainly in the UserController.

#### Registration:



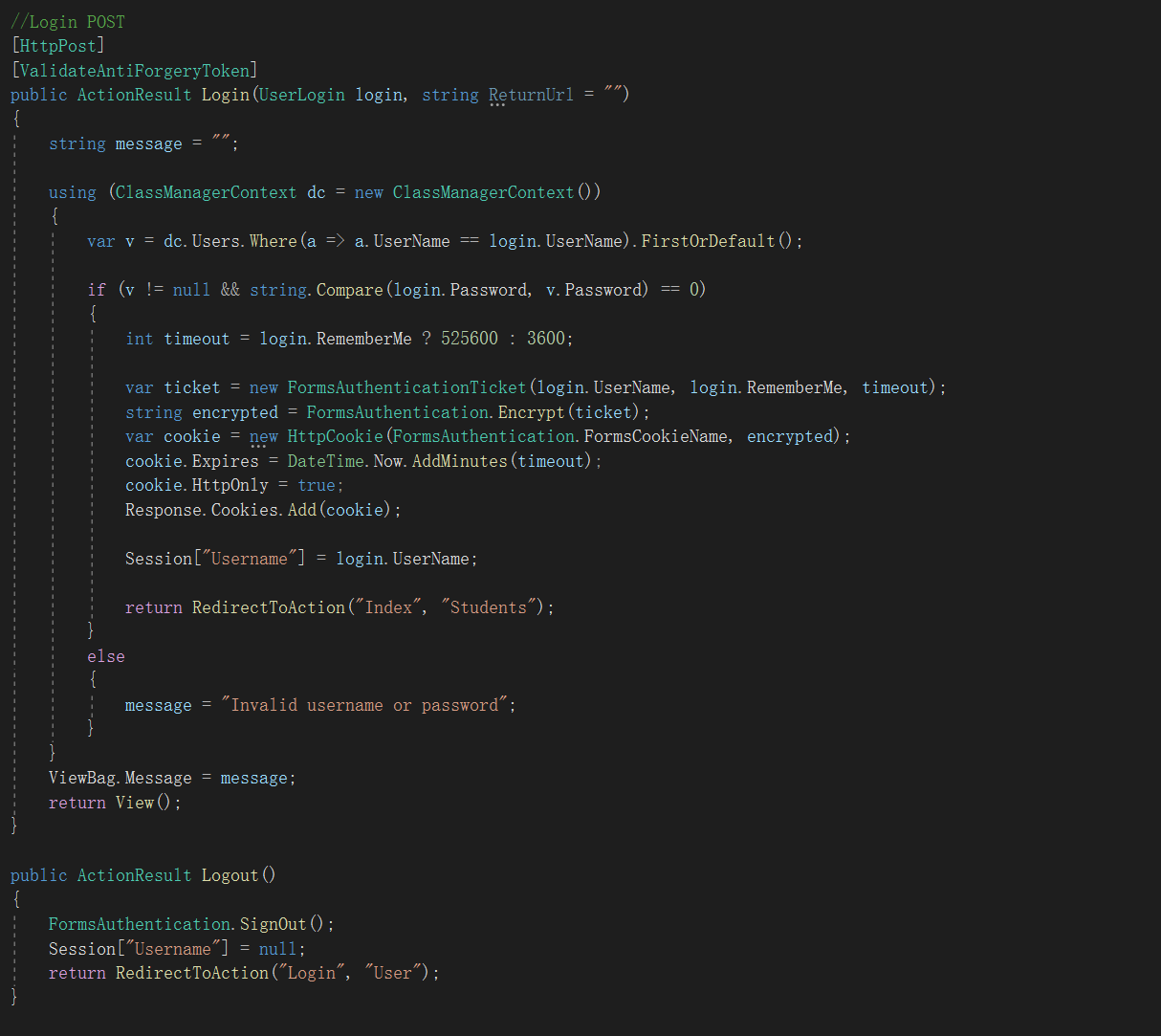
This function is responsible for handling the registration of a new user. It can only be called using an HTTP POST request. It is also decorated with the ValidateAntiForgeryToken attribute, which helps to protect against cross-site request forgery (CSRF) attacks.

The function takes a single parameter of type User, which represents the user that is being registered. The function first checks to see if the model state is valid. If the model state is valid, then the function checks to see if the username that the user has entered already exists. If the username already exists, then the function sets a message indicating that the username is already in use. Otherwise, the function adds the user to the database and sets a message indicating that the registration was successful.

If the model state is not valid, then the function sets a message indicating that the request is invalid.

Finally, the function sets the ViewBag properties Message and Status to the appropriate values and returns the View object associated with the Registration action.

#### Login:



This function is responsible for handling the login of a user. It takes a single parameter of type UserLogin, which contains the user's login credentials.

The function first checks to see if the user's credentials are valid. To do this, it queries the database for a user with the specified username and password. If a user is found, then the function checks to see if the user's RememberMe property is set to true. If it is, then the function creates a Forms authentication ticket with a longer expiration time. Otherwise, the function creates a Forms authentication ticket with a shorter expiration time.

Once the Forms authentication ticket has been created, the function encrypts it and stores it in a cookie. The function then adds the cookie to the user's browser and redirects the user to the Index action of the Students controller.

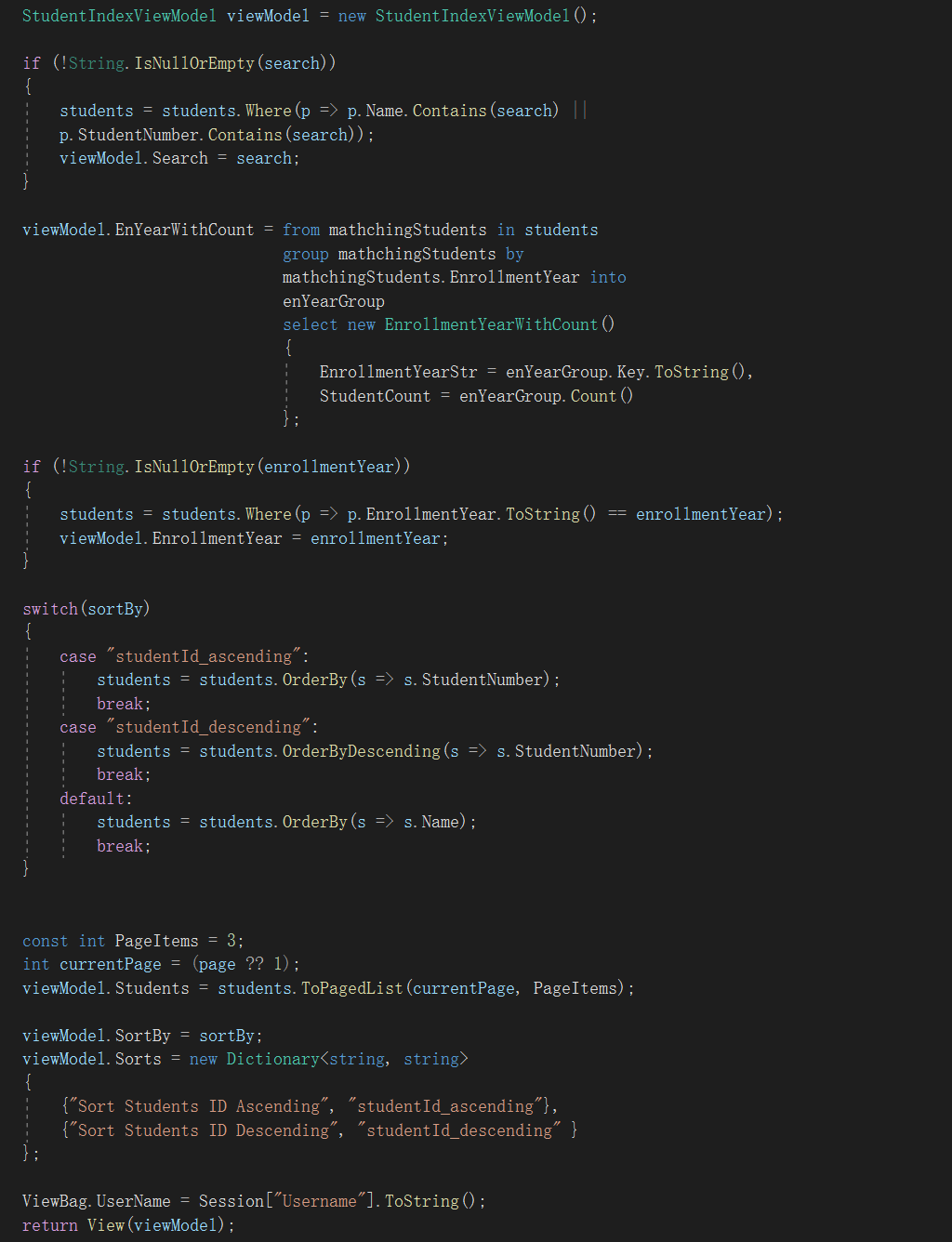
If the user's credentials are not valid, then the function sets a message indicating that the username or password is invalid and returns the View object associated with the Login action.

### Management

#### Adding, modifying, viewing, and deleting:

Entity Framework automatically generated these features, and I only made a minor change to them (about the TeacherID).

#### Search, filter, and sort:



The function performs search, filtering, and sorting using LINQ.

#### Pagination:

Pagination is implemented in the project using PagedList.Mvc.

Index.cshtml:

