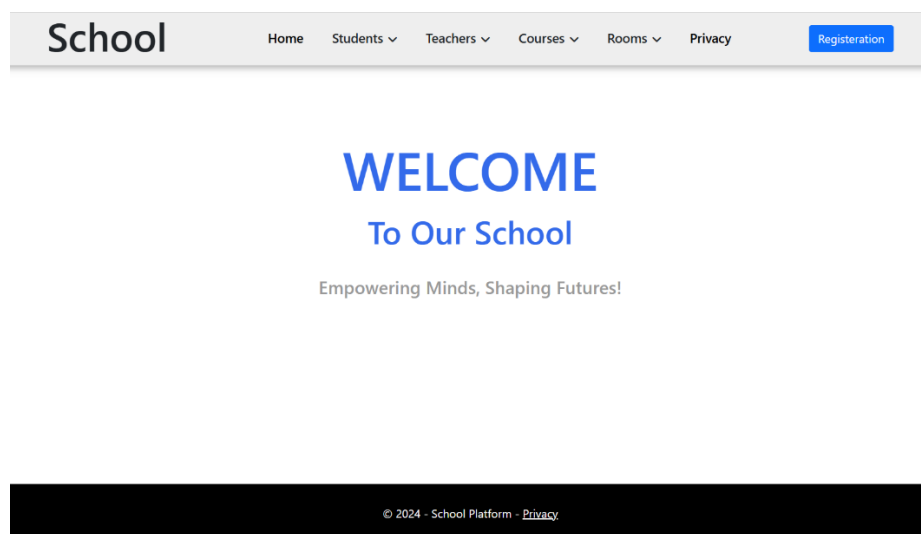


School Management System - Backend Documentation

Introduction:

The School Management System is a web-based application built using the ASP.NET MVC framework. It facilitates the management of students, courses, teachers, and classrooms. Additionally, students can register for courses and upload their photos. This documentation outlines the backend implementation, including the controllers, models, and views for each feature.



Technology Stack:

Framework: ASP.NET Core MVC

Database: SQL Server

Language: C#

ORM: Entity Framework Core

Architecture: MVC (Model-View-Controller)

System Features:

1. Students Management

Add Student: Add new students with relevant details (e.g., name, age, class).

View Students List: Display a list of all students.

Delete Student: Remove students from the system.

MVC Components:

Controller: StudentsController

Create() - GET/POST for adding new students.

Index() - Displays a list of all students.

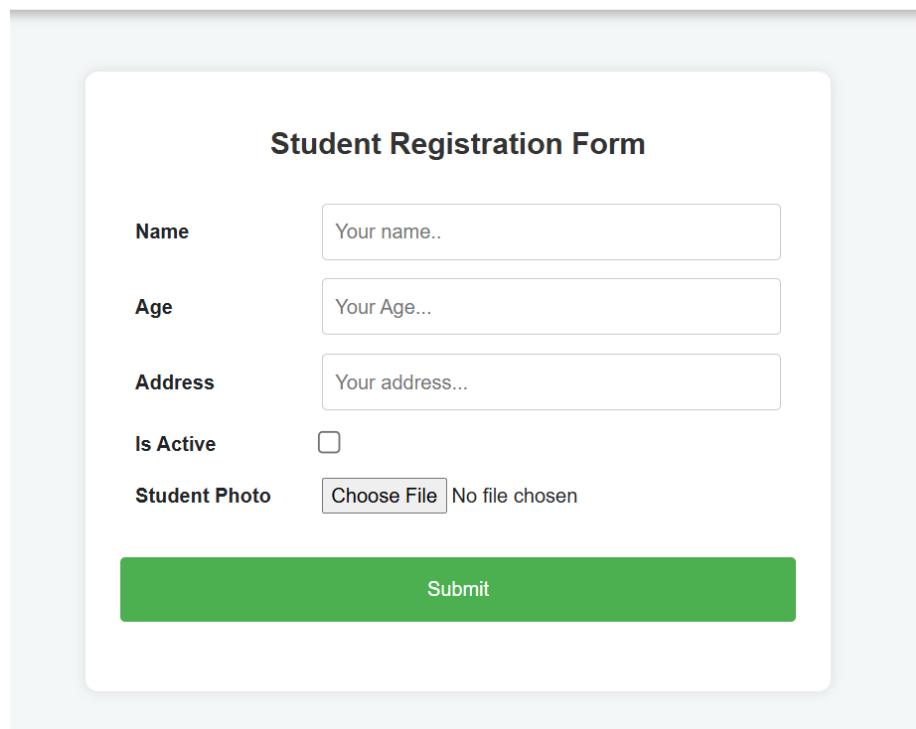
Delete(int id) - GET/POST for deleting students.

Model: Student.cs - Defines the student entity with properties such as Id, FirstName, LastName, DateOfBirth, and Class.

Views: Create.cshtml - Form to add a new student.

Index.cshtml - Displays the list of students.

Delete.cshtml - Confirm student deletion



The image shows a web form titled "Student Registration Form". It contains several input fields: "Name" with placeholder text "Your name..", "Age" with placeholder text "Your Age...", and "Address" with placeholder text "Your address...". There is a checkbox labeled "Is Active". Below that is a file upload section labeled "Student Photo" with a "Choose File" button and the text "No file chosen". At the bottom of the form is a large green "Submit" button.

2. Courses Management

Add Course: Add new courses with details like course name, status, and assigned teacher.

View Courses List: List all available courses.

Delete Course: Remove courses from the system.

MVC Components:

Controller: CoursesController

Create() - GET/POST for adding new courses.

Index() - Displays a list of all courses.

Delete(int id) - GET/POST for deleting courses.

Model: Course.cs - Defines the course entity with the following properties:

CourseName (string): The name of the course.

CourseStatus (bool): Indicates if the course is active.

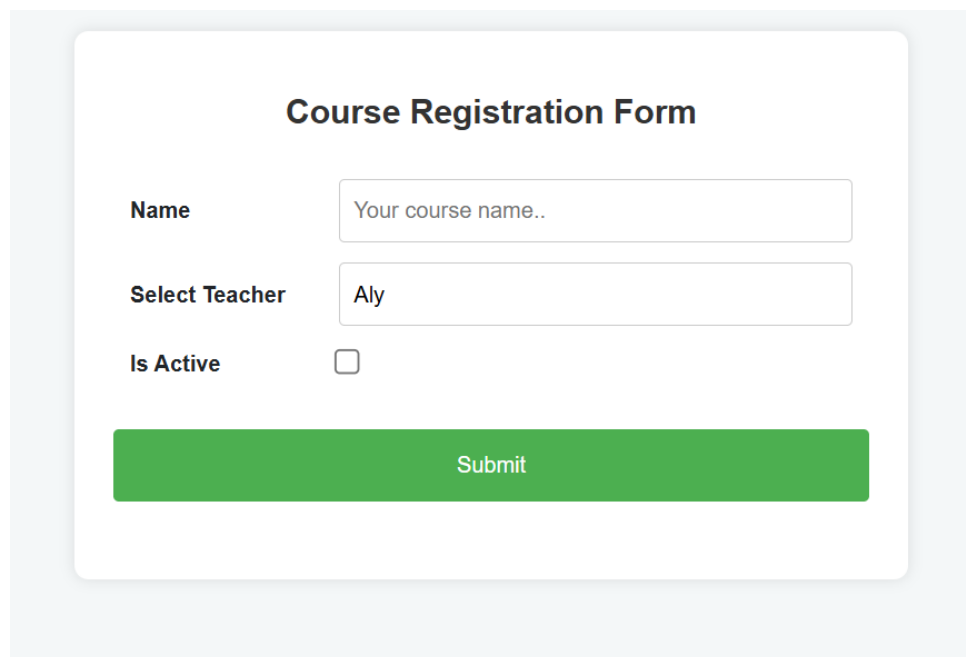
TeacherId (int): The ID of the assigned teacher.

Teacher (Teacher): A navigation property to the related Teacher entity.

Views: Create.cshtml - Form to add a new course.

Index.cshtml - Displays the list of courses.

Delete.cshtml - Confirm course deletion.



The image shows a web form titled "Course Registration Form". It contains three input fields: "Name" with a placeholder "Your course name..", "Select Teacher" with a dropdown menu showing "Aly", and "Is Active" with an unchecked checkbox. A green "Submit" button is at the bottom.

Course Registration Form

Name

Select Teacher

Is Active ☐

Submit

3. Teachers Management

Add Teacher: Add new teachers with relevant information (e.g., name, subject, age, address, status).

View Teachers List: Display all teachers.

Delete Teacher: Remove teachers from the system.

MVC Components:

Controller: TeachersController

Create() - GET/POST for adding new teachers.

Index() - Displays a list of all teachers.

Delete(int id) - GET/POST for deleting teachers.

Model: Teacher.cs - Defines the teacher entity with the following properties:

TeacherId (int): The unique ID of the teacher.

TeacherName (string): Name of the teacher (MaxLength 30).

TeacherAge (int): Age of the teacher, must be between 22 and 50 years.

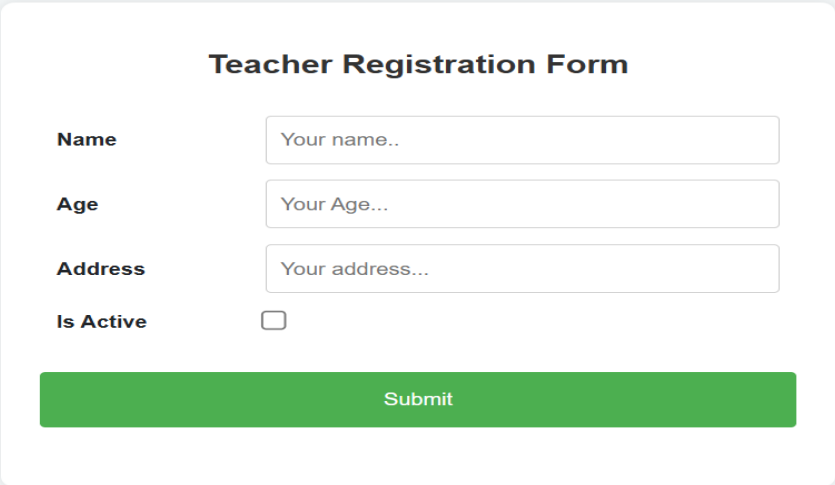
TeacherStatus (bool): Active status of the teacher.

TeacherAddress (string): Address of the teacher (MaxLength 50).

Views: Create.cshtml - Form to add a new teacher.

Index.cshtml - Displays the list of teachers.

Delete.cshtml - Confirm teacher deletion.



The image shows a web form titled "Teacher Registration Form". It contains four input fields: "Name" with placeholder text "Your name..", "Age" with placeholder text "Your Age...", "Address" with placeholder text "Your address...", and "Is Active" which is a checkbox. Below these fields is a green "Submit" button.

Teacher Registration Form	
Name	<input type="text" value="Your name.."/>
Age	<input type="text" value="Your Age..."/>
Address	<input type="text" value="Your address..."/>
Is Active	<input type="checkbox"/>
<input type="submit" value="Submit"/>	

4. Rooms Management

Add Room: Add classrooms with details like room number and capacity.

View Rooms List: Display all classrooms.

Delete Room: Remove rooms from the system.

MVC Components:

Controller: RoomsController

Create() - GET/POST for adding new rooms.

Index() - Displays a list of all rooms.

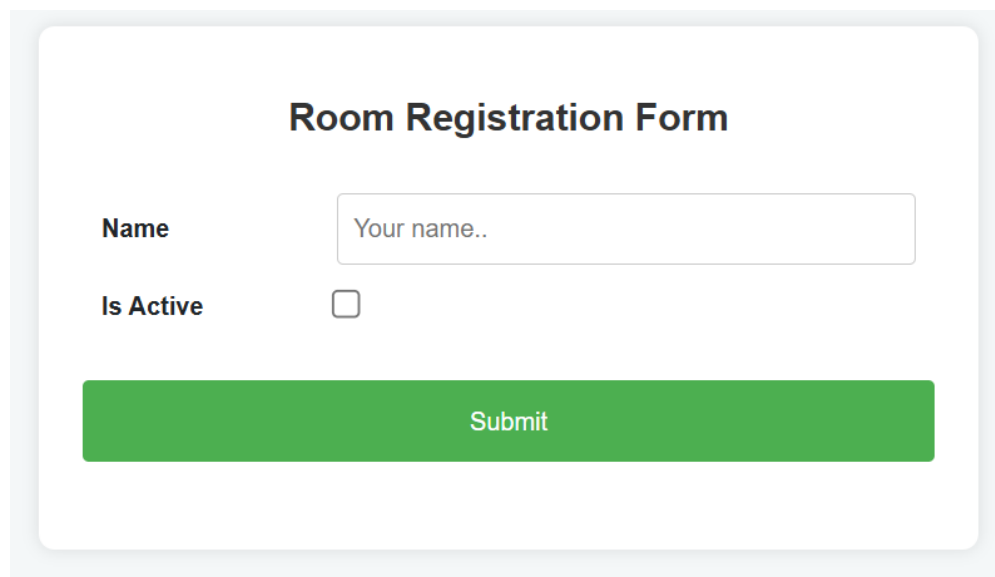
Delete(int id) - GET/POST for deleting rooms.

Model: Room.cs - Defines the room entity with properties like Id, RoomNumber, and Capacity.

Views: Create.cshtml - Form to add a new room.

Index.cshtml - Displays the list of rooms.

Delete.cshtml - Confirm room deletion.



The image shows a web form titled "Room Registration Form". It has a white background with a light gray border. The form contains two input fields: "Name" with a placeholder "Your name.." and "Is Active" with a checkbox. Below these fields is a green "Submit" button.

Room Registration Form

Name

Is Active ☐

Submit

5. Registrations

Purpose: Links students with their registered courses and allows tracking of their enrollment.

Functionality: Facilitates the registration process, allowing students to enroll in courses and view their registrations.

Database Structure:

Table Name: Registrations

Columns:

StudentId (int): The unique identifier of the student (foreign key referencing the Students table).

CourseId (int): The unique identifier of the course (foreign key referencing the Courses table).

RegistrationDate (DateTime): The date when the student registered for the course.

StudentName (string): The name of the student (derived from the Students table).

CourseName (string): The name of the course (derived from the Courses table).

MVC Components:

Controller: RegistrationsController

Create() - GET/POST to register a student for a course (select student and course).

Index(int studentId) - Display all courses the student is registered for, including student and course names.

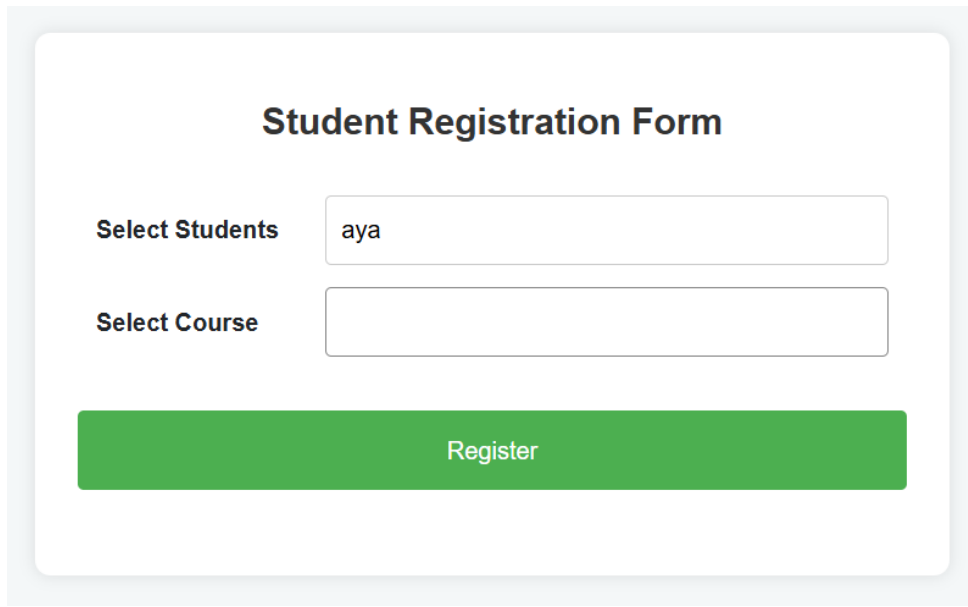
Delete(int studentId, int courseId) - Deregister a student from a course.

Model: Registration.cs - Defines the relationship between students and courses, with properties like StudentId, CourseId, and RegistrationDate.

Views: Create.cshtml - Form to register a student for a course.

Index.cshtml - Displays a list of registered courses for a student, including student and course names.

Delete.cshtml - Confirm deregistration from a course.

A screenshot of a web form titled "Student Registration Form". The form is contained within a light gray rounded rectangle. It has two input fields: "Select Students" with the text "aya" entered, and "Select Course" which is empty. Below these fields is a large green button with the text "Register" in white.

6. Add Photo to Student

Add Student Photo: Allows uploading of a profile photo for each student.

MVC Components:

Controller: PhotosController

Create(int studentId) - GET/POST to upload a photo for a student.

Index(int studentId) - Displays the uploaded photo for the student.

Model: Photo.cs - Defines the photo entity with properties like Id, StudentId, FilePath, and UploadDate.

Views:

Create.cshtml - Form to upload a student's photo.

Index.cshtml - Displays the student's uploaded photo.

File Storage:

The photo is stored on the server, and the file path is saved in the database for easy access.

Database Structure

Tables

Students: Contains student data.

Columns: Id, FirstName, LastName, DateOfBirth, Class.

Courses: Contains course data.

Columns: Id, CourseName, CourseStatus, TeacherId, Teacher.

Teachers: Contains teacher data.

Columns: TeacherId, TeacherName, TeacherAge, TeacherStatus, TeacherAddress.

Rooms: Contains room data.



Columns: Id, RoomNumber, Capacity.

Registrations: Links students with their registered courses.

Columns: StudentId, CourseId, RegistrationDate, StudentName, CourseName.

Photos: Stores student photo information.

Columns: Id, StudentId, FilePath, UploadDate.

School						
Home		Students ▾	Teachers ▾	Courses ▾	Rooms ▾	Privacy
Registration						
Name	Age	Address	Image	Status	Delete	
aya	21	egypt		False ? "Active" : "Inactive"	x	
aly	18	menof		False ? "Active" : "Inactive"	x	

Conclusion

This documentation outlines the backend structure of the School Management System using the ASP.NET MVC pattern. The system allows for the management of students, courses, teachers, rooms, a registration module for students, and a photo uploading feature. The system does not currently include editing functionality for students or courses.