A

SYNOPSIS

of

MINOR PROJECT

on

Collage Result Management

Submitted by
Yuvraj Singh Chouhan (21EGICS126)

Project guide
Ms. Kanika garg

Head of Department
Dr. Mayank Patel

Abstract :

- The project titled “COLLAGE RESULT MANAGEMENT SYSTEM” is a web based application developed to maintain the results of the students.

This project contains three modules. Register/login module, admin module and student module. Admin module is used to create and manage the subjects ,classes and Add the students and their results. Student module used to provide the results of the students and can download the results.

- Registered users can access the database of the students.

which will provide details about subjects, classes, results, And student can download results .which is useful for the students.

EXISTING SYSTEM

- Existing process of the getting information for the student is done manually.
- All the wanted information need to be provided by the administration.
- If the students want to know the results of the particular subject so their should wait for long time because work done manually.
- The information about their courses , results and other information accessing is not possible.

PROPOSED SYSTEM

We have successfully proposed the “COLLAGE RESULT MANAGEMENT SYSTEM” for replacing the manual work of the administration.

- By this application student can easily access the modules like student results and courses other information required to student.
- This application is flexible and can easily access by the student .
- So the time taken for getting the information will be reduces.

SYSTEM REQUIREMENT

- Hardware Requirement and
- Software Requirement

SOFTWARE REQUIREMENTS

- Front-End Design : HTML , CSS, JS, **Bootstrap**
- Back-End Database : Microsoft SQL Server, PHP

Modules

- 1.Student Module
- 2.Admin Module

Student Module

2. Student Module:

- students can view the results.
- student can download the results..

Admin Module

3. Admin Module:

- Create and manage the subjects , classes and can add the students and their results of the students.
- Reports for various modules

Update or deleting the admin information

Creating a College Management System using ASP.NET involves multiple steps and various components. Here's a basic example to get you started.
This example will cover a simple setup, including student registration, course management, and student enrollment. For a more comprehensive system, you would expand of these concepts.

Create a New ASP.NET Web Application:

Open Visual Studio and create a new ASP.NET Web Application.

Add Models:

First, create the models for the system. These models will represent the data structures.

```
// Models/Student.cs
public class Student
{
    public int StudentID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime EnrollmentDate { get; set; }

    public virtual ICollection<Enrollment> Enrollments { get; set; }
}

// Models/Course.cs
public class Course
{
    public int CourseID { get; set; }
    public string Title { get; set; }
    public int Credits { get; set; }

    public virtual ICollection<Enrollment> Enrollments { get; set; }
}

// Models/Enrollment.cs
public class Enrollment
{
    public int EnrollmentID { get; set; }
    public int CourseID { get; set; }
    public int StudentID { get; set; }
    public Grade? Grade { get; set; }

    public virtual Course Course { get; set; }
    public virtual Student Student { get; set; }
}

public enum Grade
{
    A, B, C, D, F
}
```

Add the Database Context:
Create a new class for the database context.

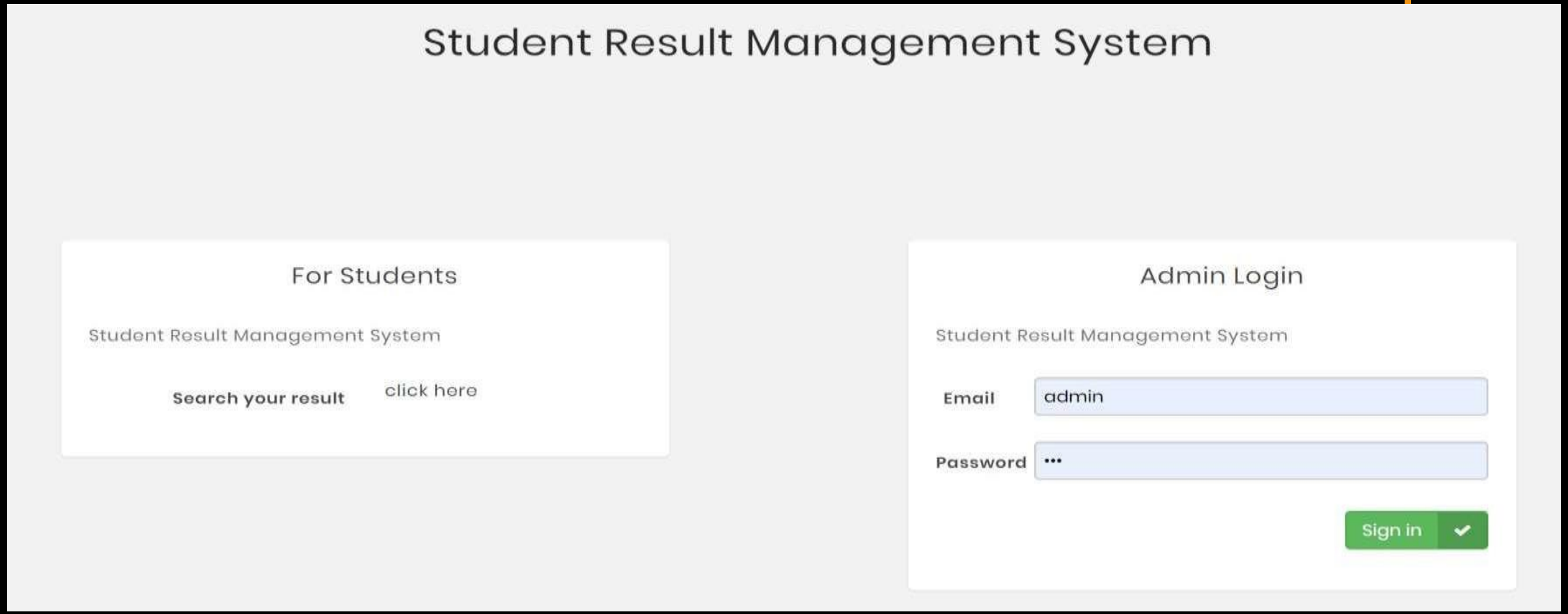
```
// Data/SchoolContext.cs
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;

public class SchoolContext : DbContext
{
    public SchoolContext() : base("SchoolContext")
    {
    }

    public DbSet<Student> Students { get; set; }
    public DbSet<Course> Courses { get; set; }
    public DbSet<Enrollment> Enrollments { get; set; }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
    }
}
```

Create the Controllers:
Create controllers to handle HTTP requests.

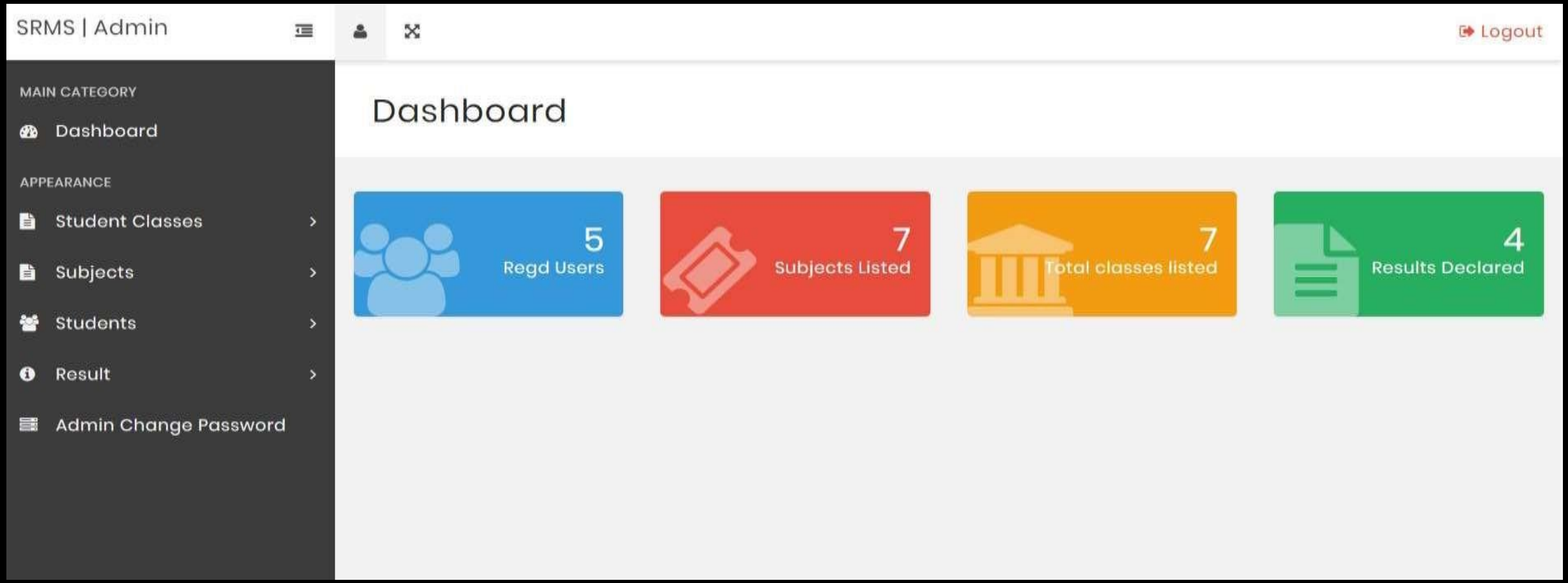
```
// Controllers/StudentsController.cs
using System.Data.Entity;
using System.Net;
using System.Threading.Tasks;
using System.Web.Mvc;

public class StudentsController : Controller
{
    private SchoolContext db = new SchoolContext();

    public async Task<ActionResult> Index()
    {
        return View(await db.Students.ToListAsync());
    }

    public async Task<ActionResult> Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Student student = await db.Students.FindAsync(id);
        if (student == null)
        {
            return HttpNotFound();
        }
        return View(student);
    }

    public ActionResult Create()
    {
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult> Create([Bind(Include = "StudentID,FirstName,LastName,EnrollmentDate")] Student student)
    {
        if (ModelState.IsValid)
        {
            db.Students.Add(student);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }

        return View(student);
    }

    // Other actions (Edit, Delete) can be implemented similarly
}

// Controllers/CoursesController.cs
using System.Data.Entity;
using System.Net;
using System.Threading.Tasks;
using System.Web.Mvc;
```

```
public class EnrollmentsController : Controller
{
    private SchoolContext db = new SchoolContext();

    public async Task<ActionResult> Index()
    {
        var enrollments = db.Enrollments.Include(e => e.Course).Include(e => e.Student);
        return View(await enrollments.ToListAsync());
    }

    public async Task<ActionResult> Details(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Enrollment enrollment = await db.Enrollments.FindAsync(id);
        if (enrollment == null)
        {
            return HttpNotFound();
        }
        return View(enrollment);
    }

    public ActionResult Create()
    {
        ViewBag.CourseID = new SelectList(db.Courses, "CourseID", "Title");
        ViewBag.StudentID = new SelectList(db.Students, "StudentID", "FullName");
        return View();
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult> Create([Bind(Include = "EnrollmentID,CourseID,StudentID,Grade")] Enrollment enrollment)
    {
        if (ModelState.IsValid)
        {
            db.Enrollments.Add(enrollment);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }

        ViewBag.CourseID = new SelectList(db.Courses, "CourseID", "Title", enrollment.CourseID);
        ViewBag.StudentID = new SelectList(db.Students, "StudentID", "FullName", enrollment.StudentID);
        return View(enrollment);
    }

    // Other actions (Edit, Delete) can be implemented similarly
}
```

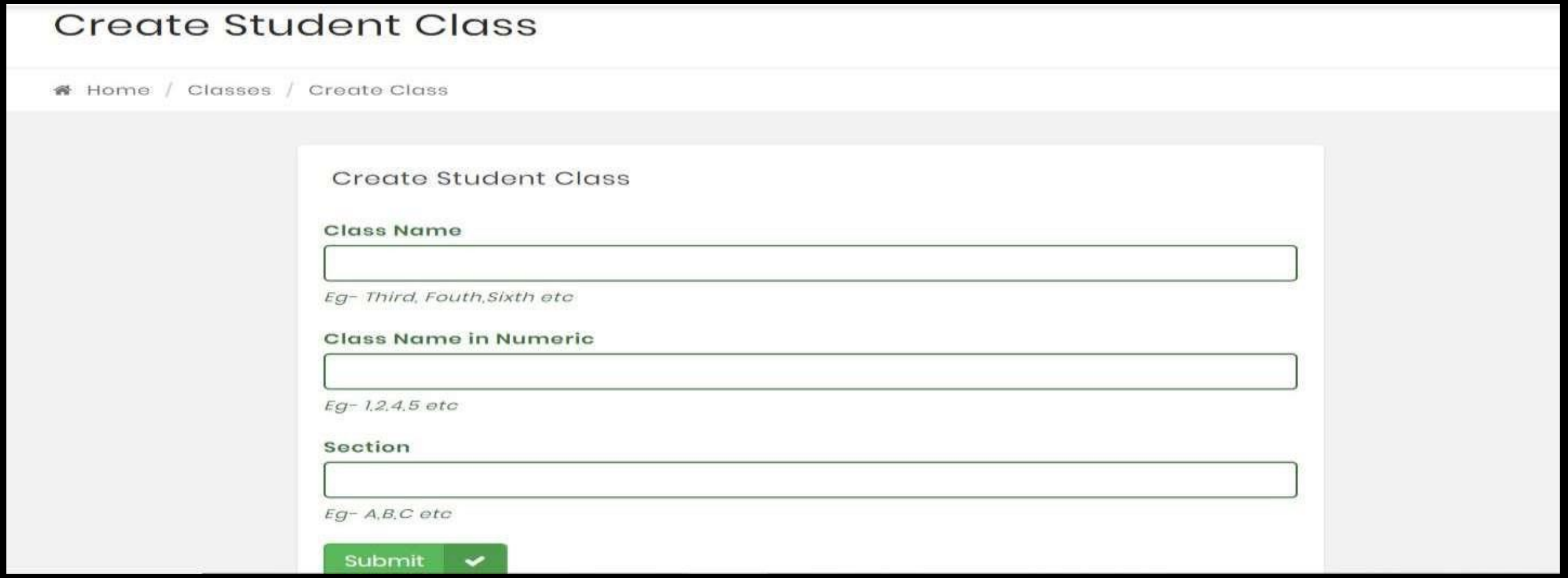
index


Student Result Management System
For Students
Student Result Management System
Search your result
click here
Admin Login
Student Result Management System
Email
admin
Password
Sign in

Dashboard

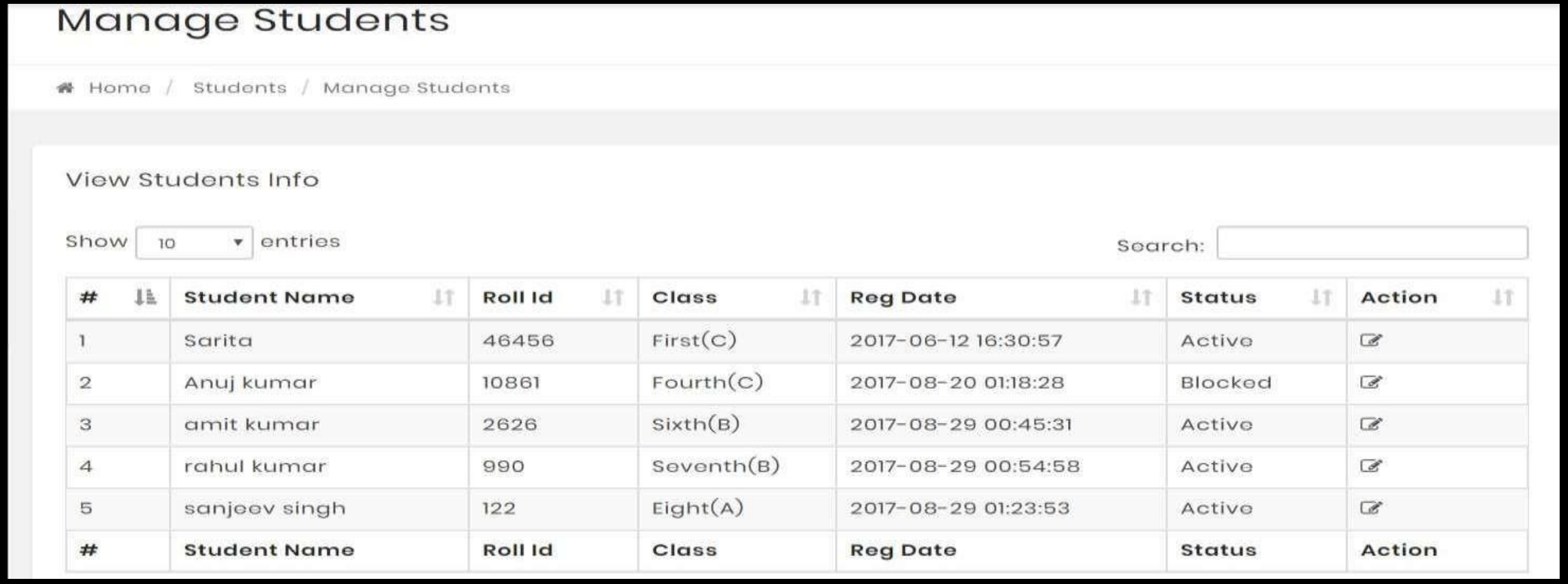

SRMS | Admin
Logout
MAIN CATEGORY
Dashboard
APPEARANCE
Student Classes
Subjects
Students
Result
Admin Change Password
Dashboard
5
Regd Users
7
Subjects Listed
7
Total classes listed
4
Results Declared

Student Create



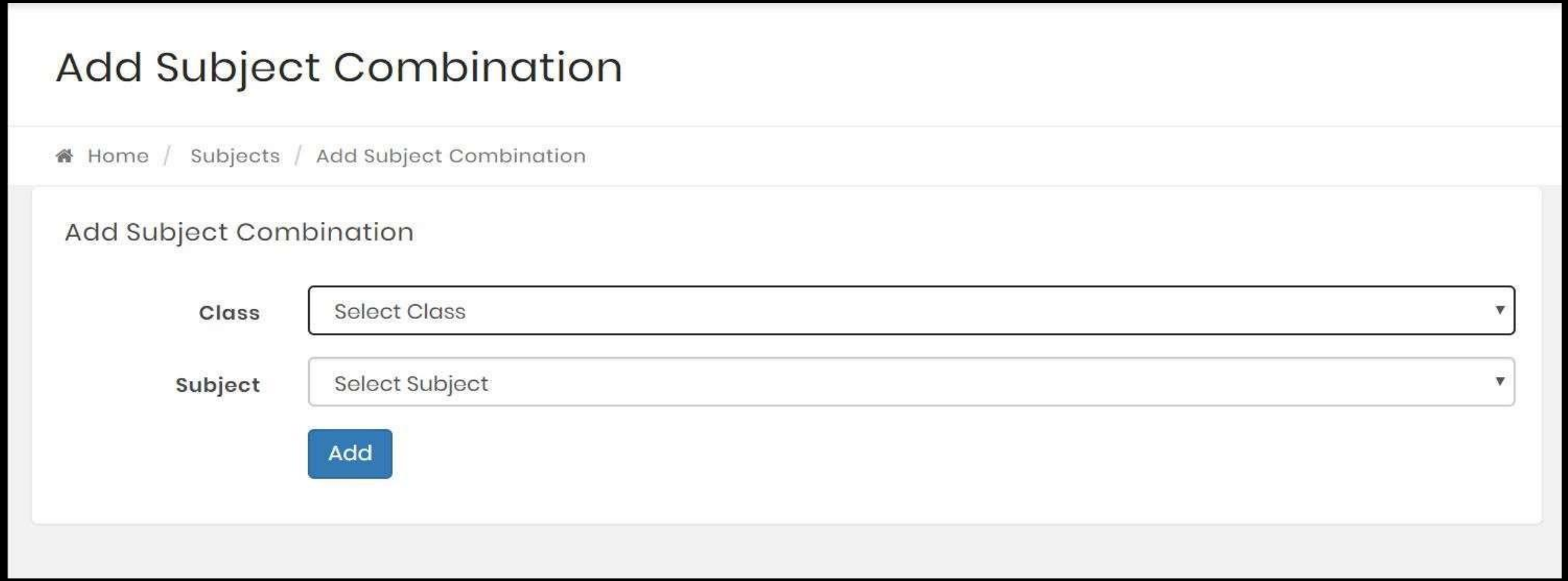
Student Admission

Manage Student

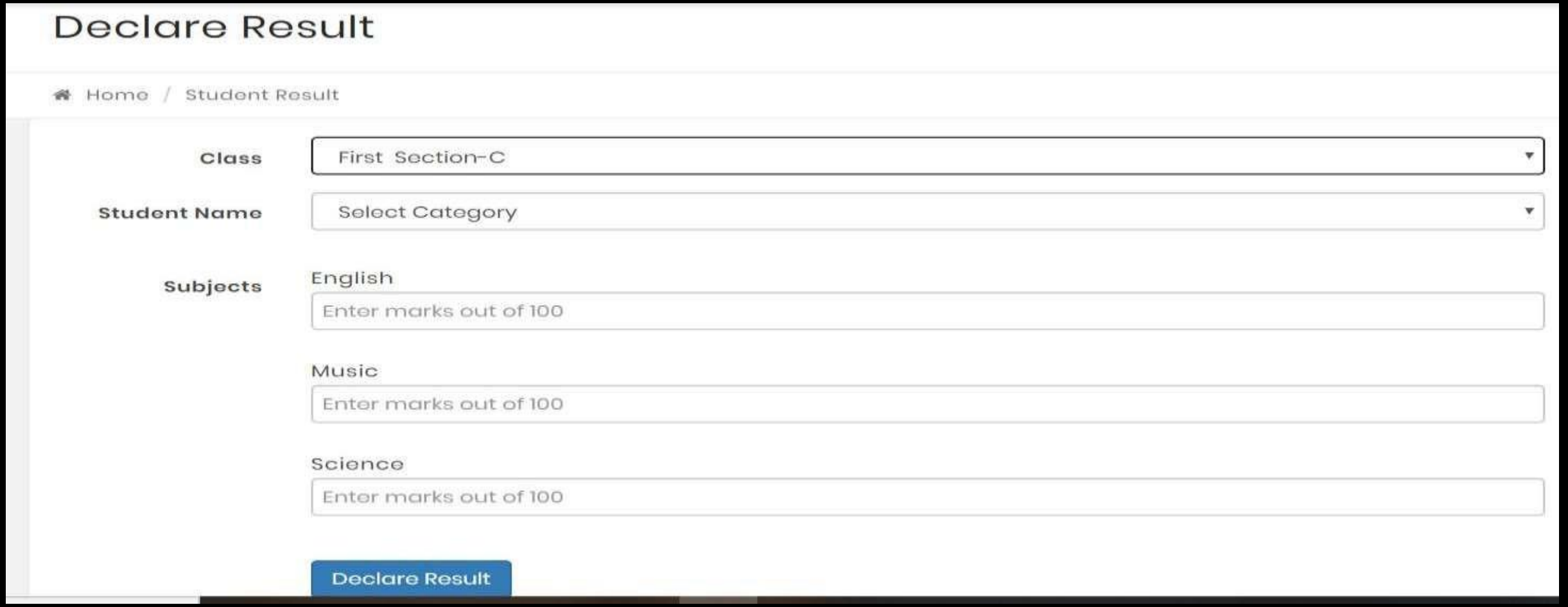


Subject Create

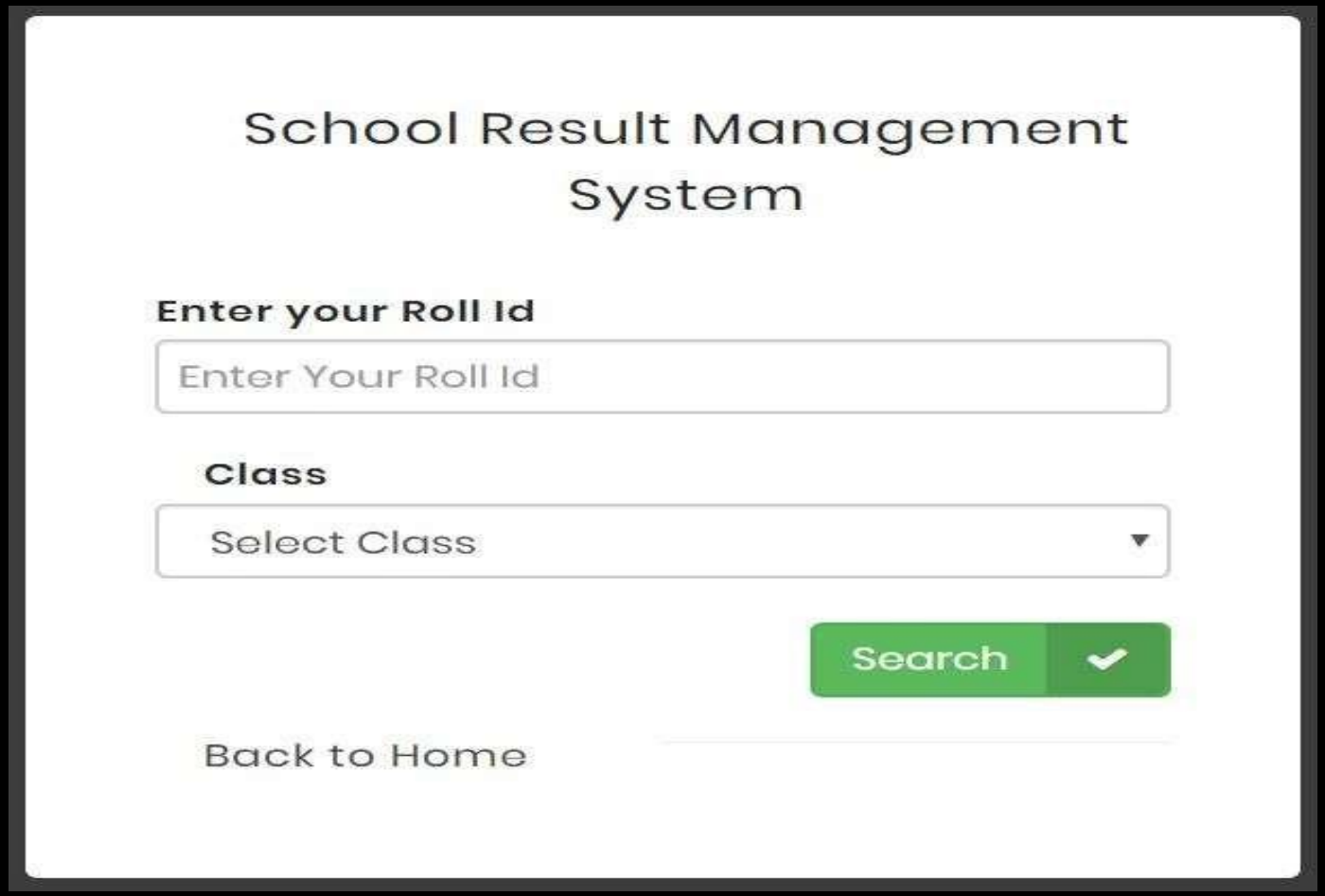
Subject Combination



Result Declare



Result Search



Student Result Details

Student Result Details

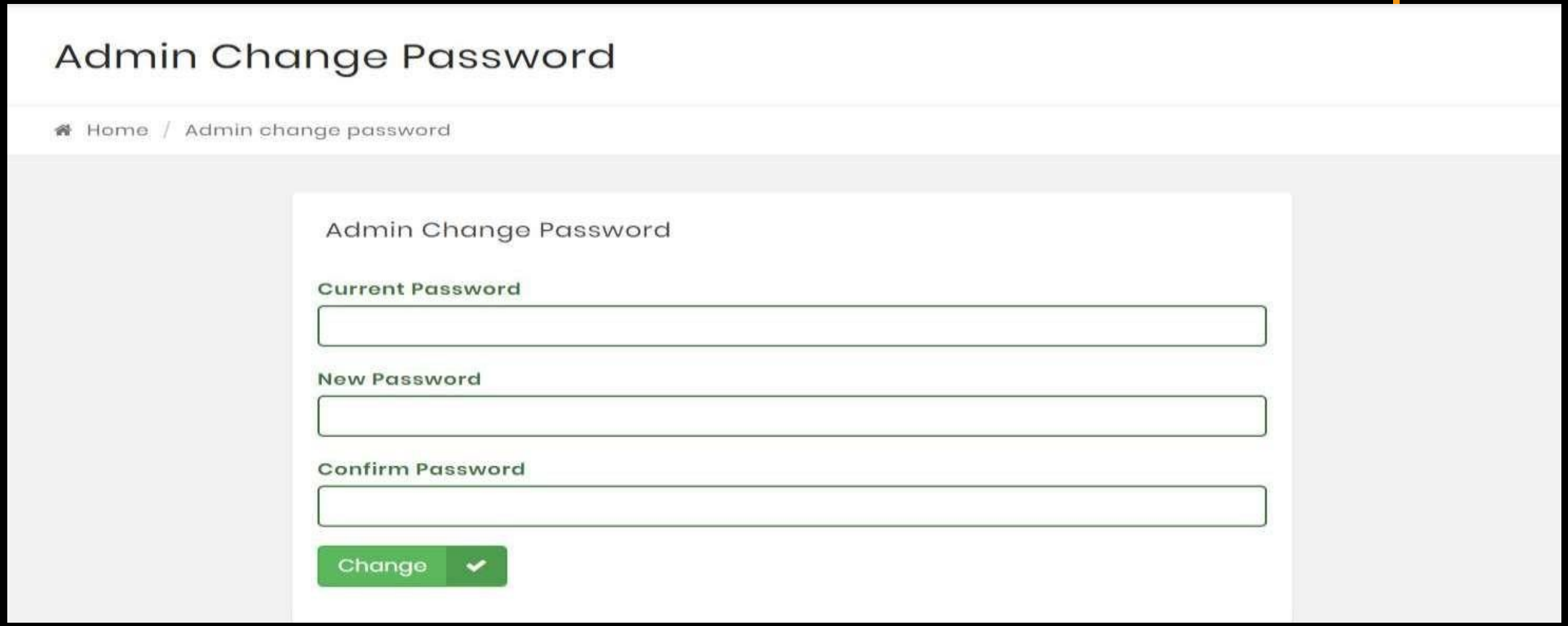
Student Name : Anuj kumar

Student Roll Id : 10861

Student Class: Fourth(C)

#	Subject	Marks
1	English	90
2	Maths	75
3	Music	56
4	Science	80
Total Marks		**301** out of **400**
Percntage		**75.25 %**

Admin Change Password



THANK YOU