Marking Scheme

Informatics College Pokhara

Application Development
CS6004NI
Course Work 1

Submitted By: **Sirish Babu Acharya**
London Met ID: Enter ID Here

Submitted To: Ishwor Sapkota
Module Leader

Component Grade and Comments	
A. Implementation of Application	
User Interface and proper controls used for designing	User Interface is complete but not separated and have proper use of controls
Manual data entry or import from csv	not properly saved or imported data
Data Validation	Only basic validation
Enrollment Report & weekly report in tabular format	very poorly executed reports and data not shown accurately
Course wise enrollment report & Chart display	Very poorly designed and only contains one report format with in appropriate data
Algorithm used for sorting & proper sorting of data	Default sorting provided by .net is used
B. Documentation	
User Manual for running the application	User Manual is below average. Is textual only.

Application architecture & description of the classes ad methods sued	average work with very limited explanation of the classes and methods used
Flow chart, algoriathms and data sctructures used	average work with very limited explanation and missing diagramatic representation.
Reflective essay	Very poorly written
C. Programming Style	
Clarity of code,Popper Naming convention & comments	very poorly written code and no comments at all
System Usability	very poorly developed application

Overall Grade:	**E+**	**E+**

Overall Comment:

Code should be self explainable with less comments. Need some proper naming of the component and require to add comments on required area.

In overall the code is working and all the functionality seems working and system can be used

Module Code & Module Title

CU6004NP Advanced Database

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Name: Srish Babu Acharya

College ID: NP04CP4A170062

University ID: 17030534

Abstract

It is an individual coursework of a subject called Application Development. The coursework is about developing a software system that allows to keep record of students enrolled in a college. The software is developed in the visual studio using a C# programming language. The project was assigned in the 5th week while the submission is in the 11th week.

The main idea of the project is to keep entry and record of enrolled student in college or a school. The admin can generate a report in CSV format and export it so that viewers can see through excel without any problem or complication. The project allows to view weekly records of students.

Table of Contents

List of Figures

Introduction

This project is all about the Student management system. The system has full functionality and is capable of working as the design shows. It does everything that is required to record data of student in a school/college. The system has login page and landing home page where raw data are saved. It can make weekly report of enrolled students. Data can also be sorted by the first name in an alphabetical order and can sort by registration date as well. There are many more functionalities of the program and are mentioned in the full documentation.

Current scenario

Though it is a technological era, many companies do not use a digital system to store data. The system they use is completely outdated. Companies are not able to grasp the technology and handle them so. They are still not successful to make full use of digital system. To alter, digital systems need to be rooted more in the education sector.

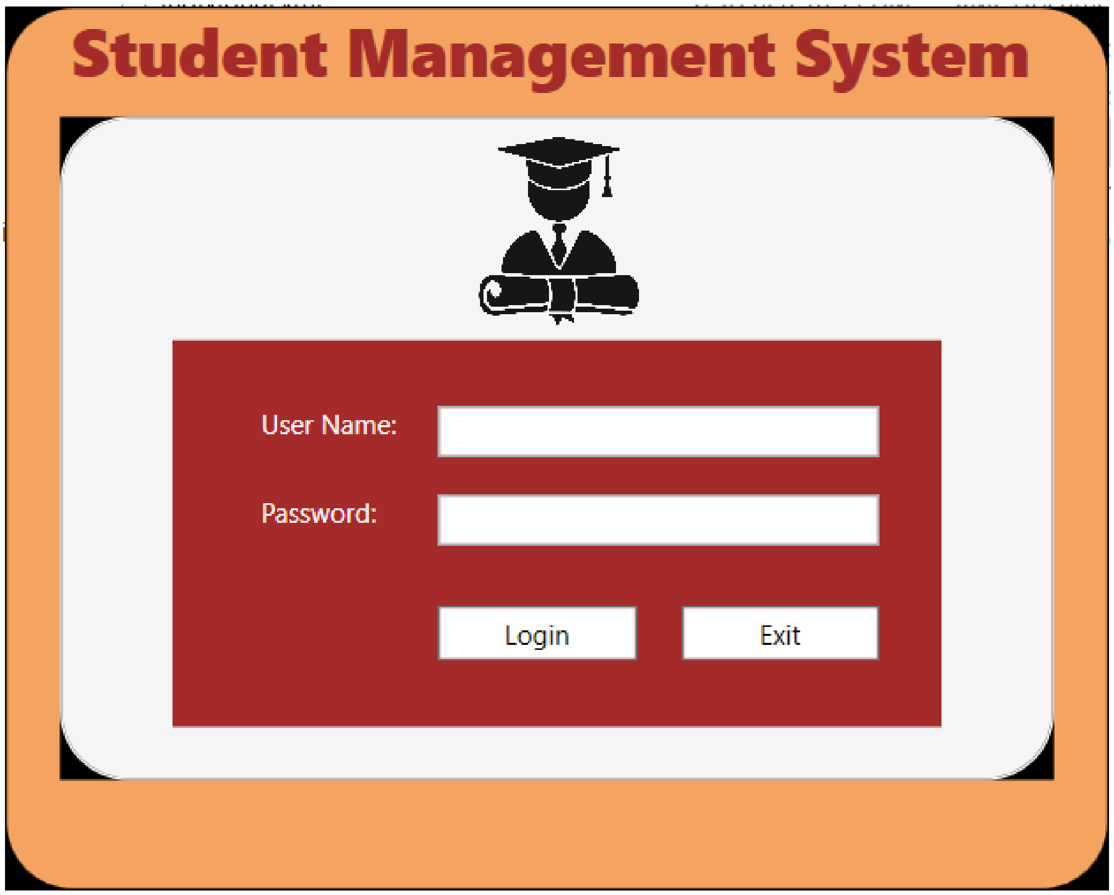
Project Description

This software is perfect for the schools and colleges. The sole feature of this software system is to store data of students like their name, number, address, enrolled course, registration date and as well as the registration id. After adding the store details the details are saved/export as the csv file and that of csv file will be used later to display student details and to make weekly report. Data transaction can be done on daily and weekly basis.

User Manual

There are screenshots below to show how to operate the system.

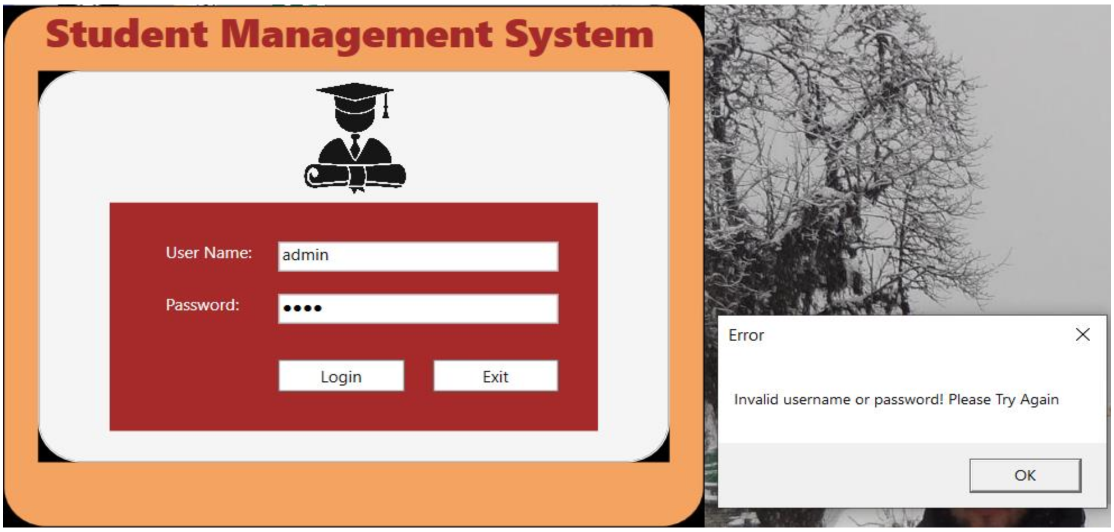
1. After opening the system, login page will appear.



Student Management System

Figure 1: Login Page UI

2. Valid username and password is set to the login page to give access to the entire system. If the username and password are incorrect then dialog box will be appeared showing error message.



Invalid username or password! Please Try Again

Figure 2: login page with dialog message for invalid username or password

3. The username and password are set as “admin”. Valid username and password will only give access to the system. After entering the valid username and password the landing page will be appeared as follows. Now the user can access other features available in the system such as enrolling student and so on.

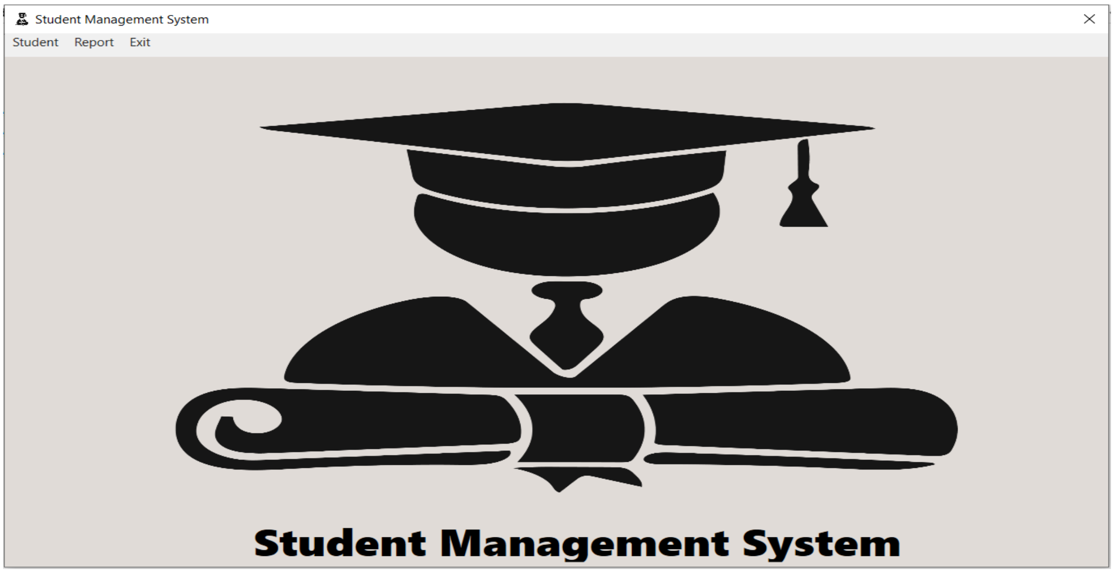


Figure 3: Homepage after entering valid username and password

Architecture

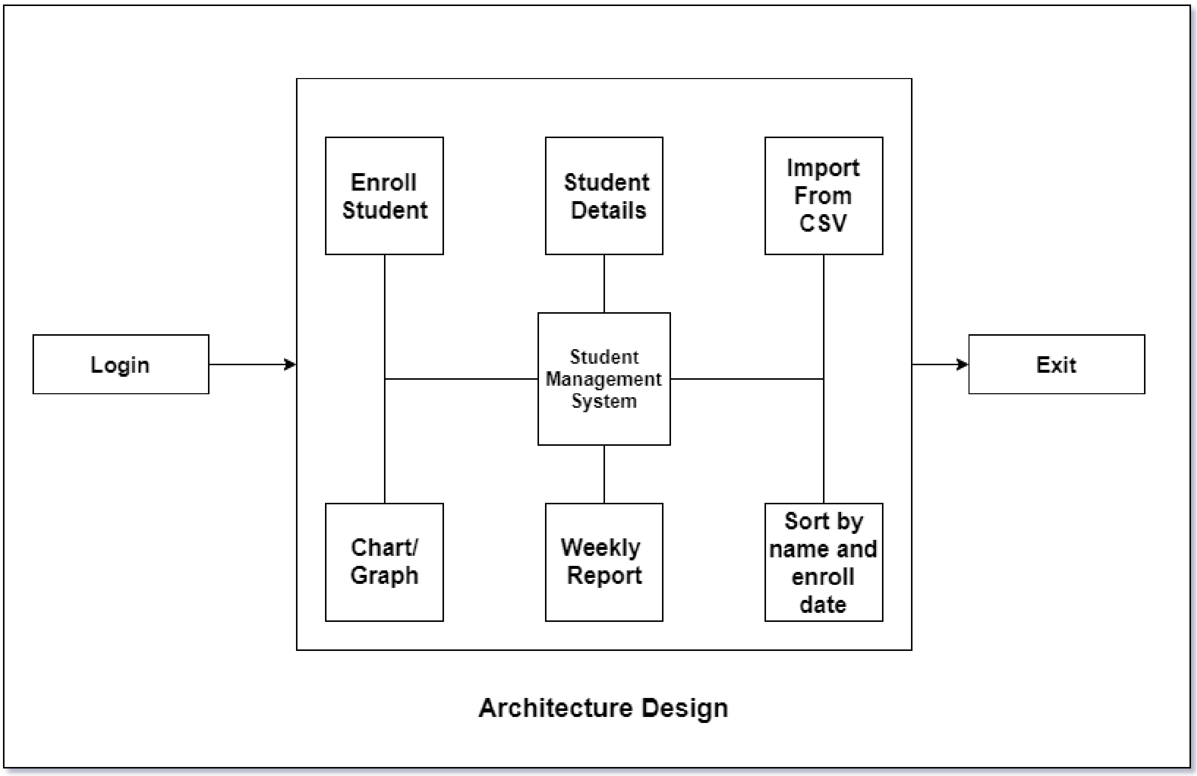


Figure 4: Architecture design diagram of the system

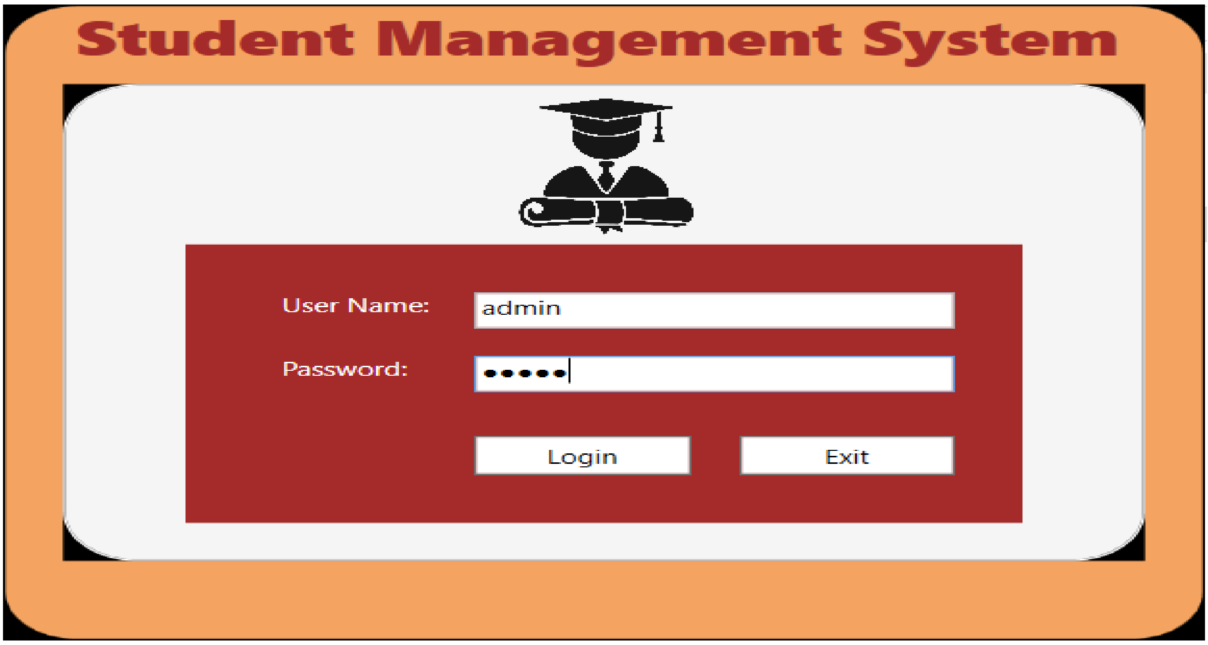
The system requires a valid username and password to login. The user gets access to the main page only after a successful login with valid username and password. Main page holds the report and student as a menu item in the navigation bar. After clicking the enroll student menu item from the navigation bar then the student enrolling page will be appeared to add details of student. Admin can add the information and then log out of the system.

Testing

Test case 1

Objective: To check whether the login page working properly or not.

Result: successfully logged into the system with username and password admin and then home page appeared.



User Name: admin

Password: •••••

Figure 5: Login page

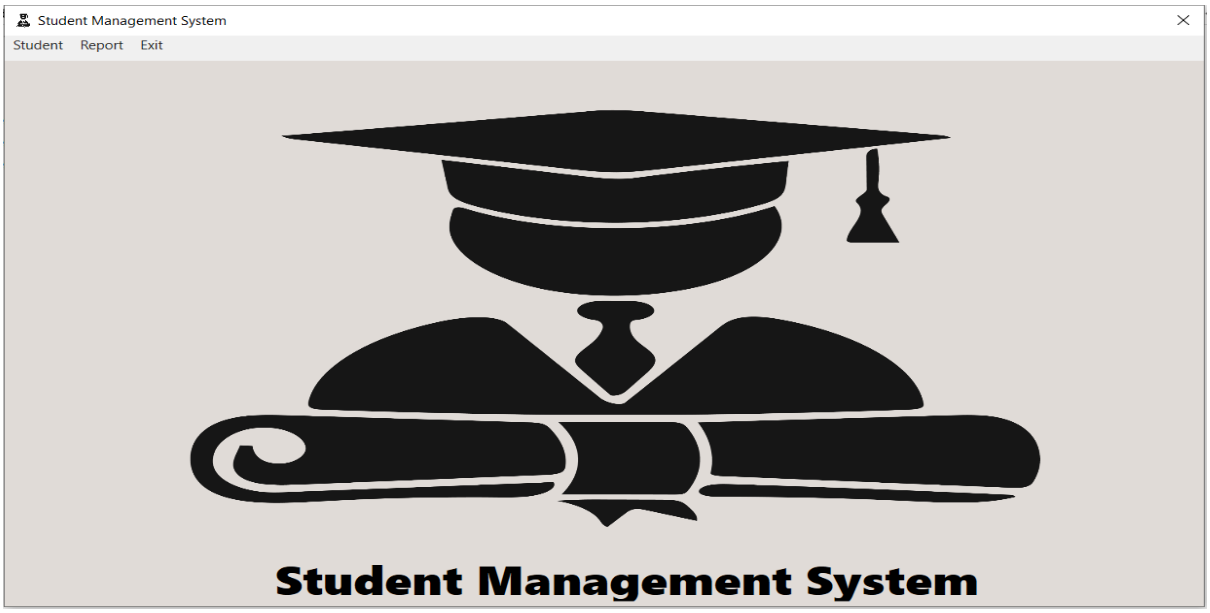


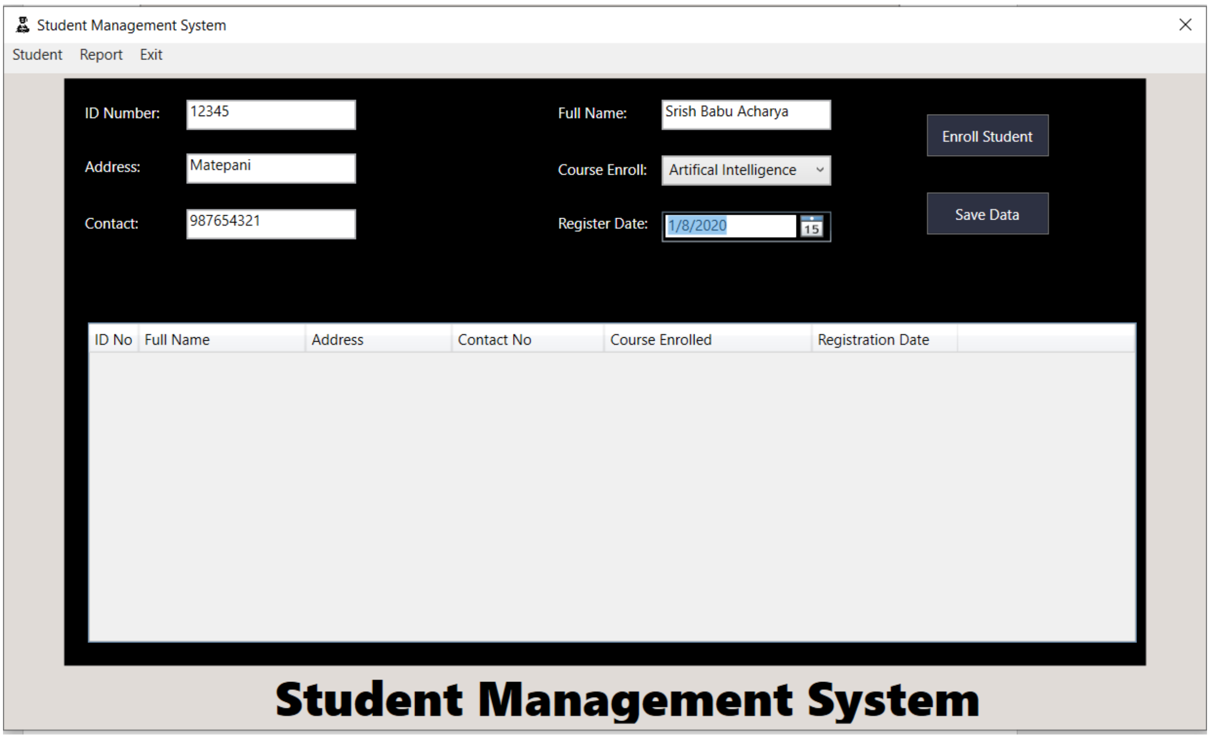
Figure 6: Homepage with menu items on navbar

Test Case 2:

Objective: To input data in the textboxes in the student enroll page and to save the data in the form of student.csv file and to display data in the datagrid below the text boxes as shown in the figure.

After clicking the menu item “Student enroll” the page as shown in below will appear to input the record of the student.

Result: Student is added successfully and exported to studentdetails.csv file and the data is also shown in the datagrid below.



ID Number: 12345

Full Name: Srish Babu Acharya

Address: Matepani

Course Enroll: Artifical Intelligence

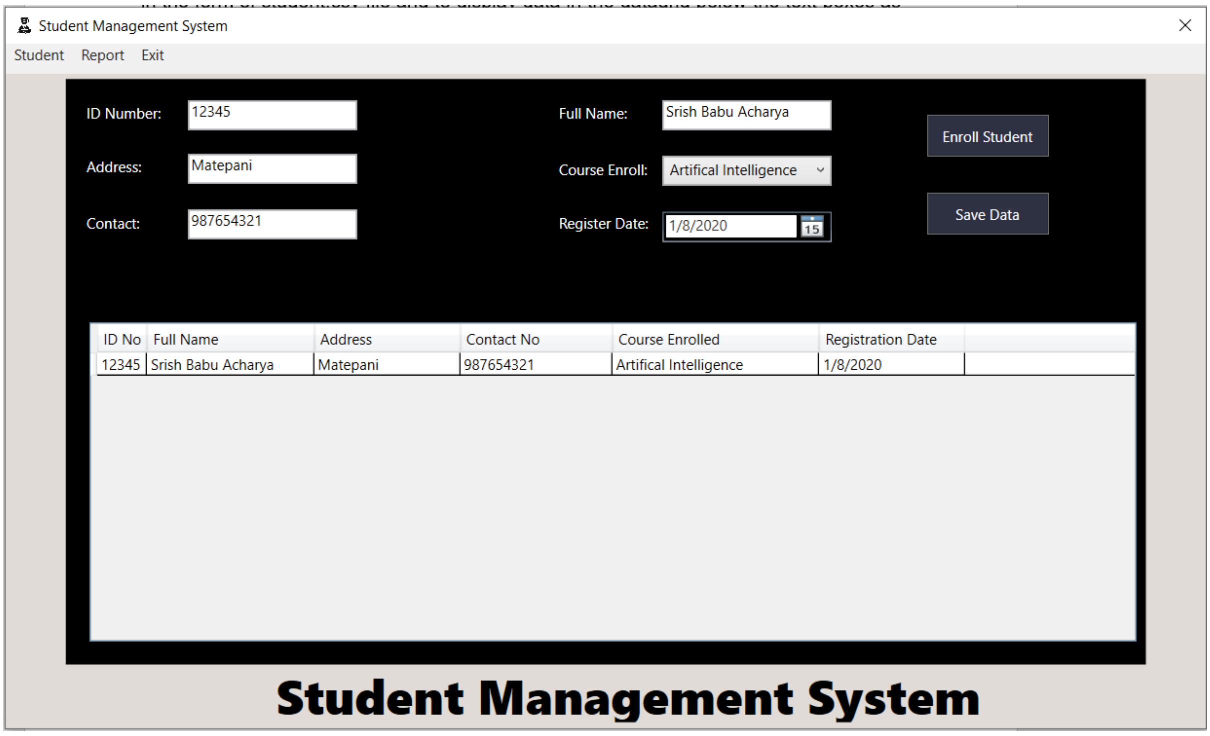
Contact: 987654321

Register Date: 1/8/2020

ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date

Student Management System

Figure 7: Data added in textfield of enroll student page

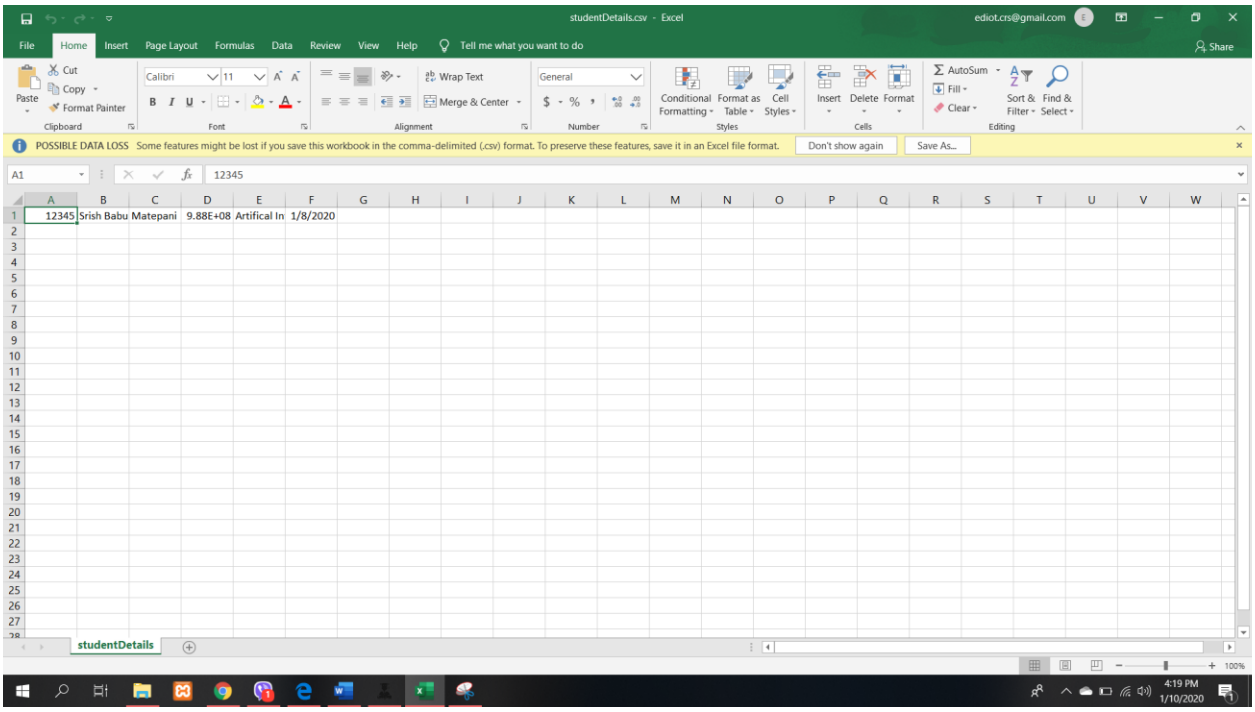


ID Number:	12345	Full Name:	Srish Babu Acharya
Address:	Matepani	Course Enroll:	Artifical Intelligence
Contact:	987654321	Register Date:	1/8/2020

ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date
12345	Srish Babu Acharya	Matepani	987654321	Artifical Intelligence	1/8/2020

Figure 8: Enrolled student details

Student details is exported as studentdetails.csv file as shown in the figure below.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

	A	B	C	D	E	F
1	12345	Srish Babu	Matepani	9.88E+08	Artifical In	1/8/2020

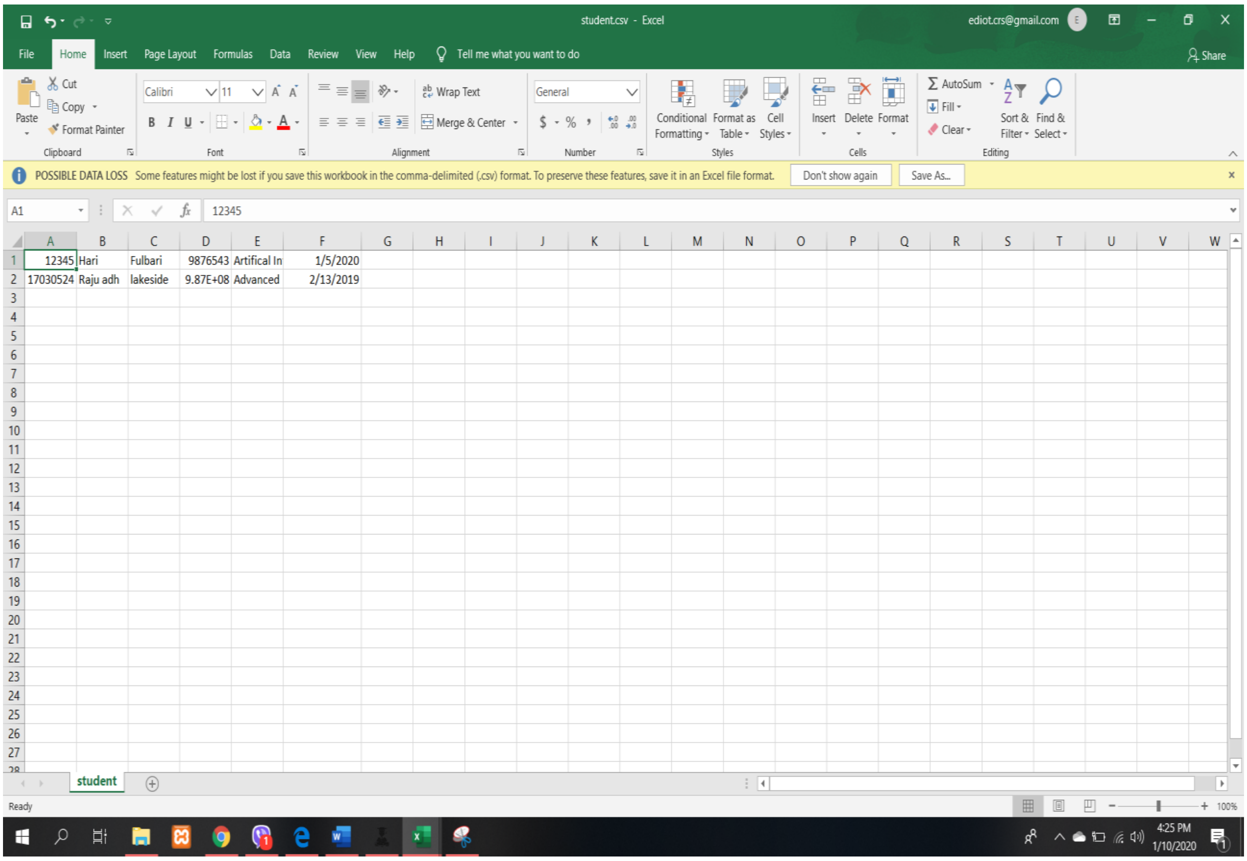
Figure 9: Studentdetails.csv file after enrolling student

Test Case 3:

Objective: To import csv file and show the details in the datagrid and again add those details into the studentdetails.csv file.

Result: imported successfully and added the data to studentdetails.csv file.

Student.csv to import into the system.

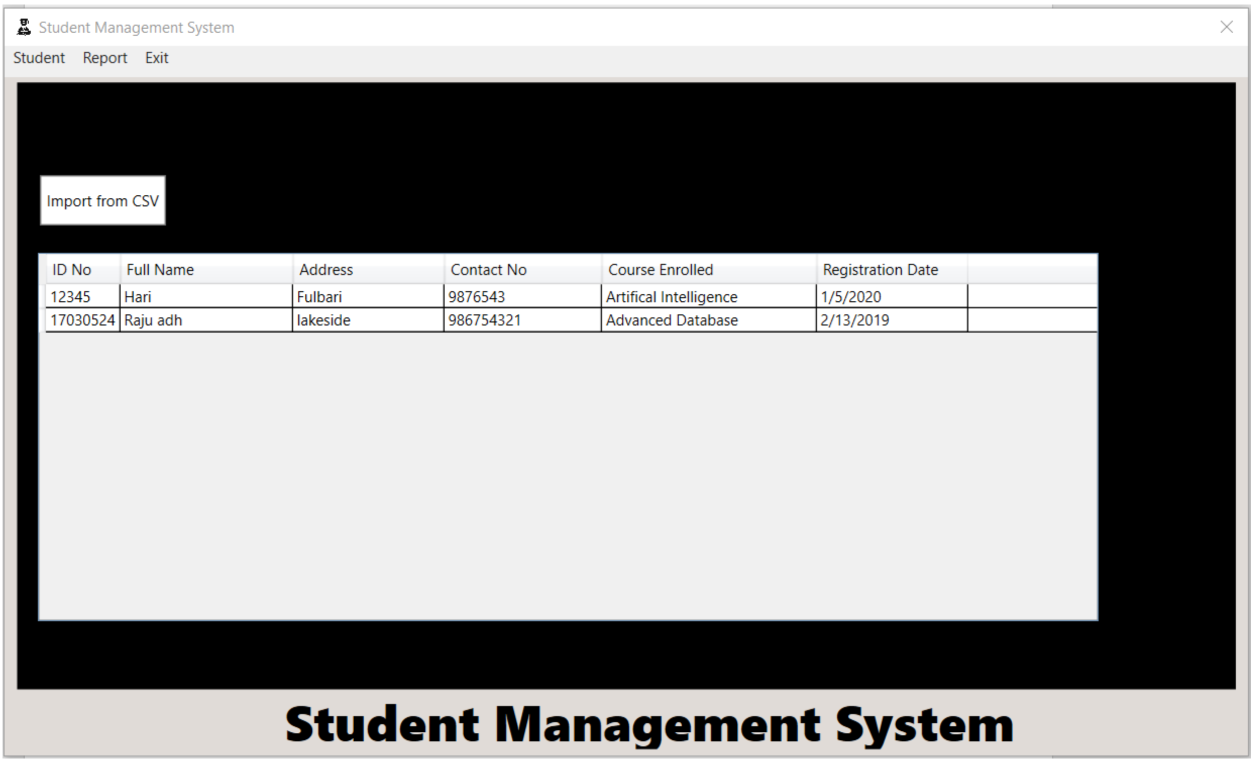


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 12345

	A	B	C	D	E	F
1	12345	Hari	Fulbari	9876543	Artifical In	1/5/2020
2	17030524	Raju adh	lakeside	9.87E+08	Advanced	2/13/2019

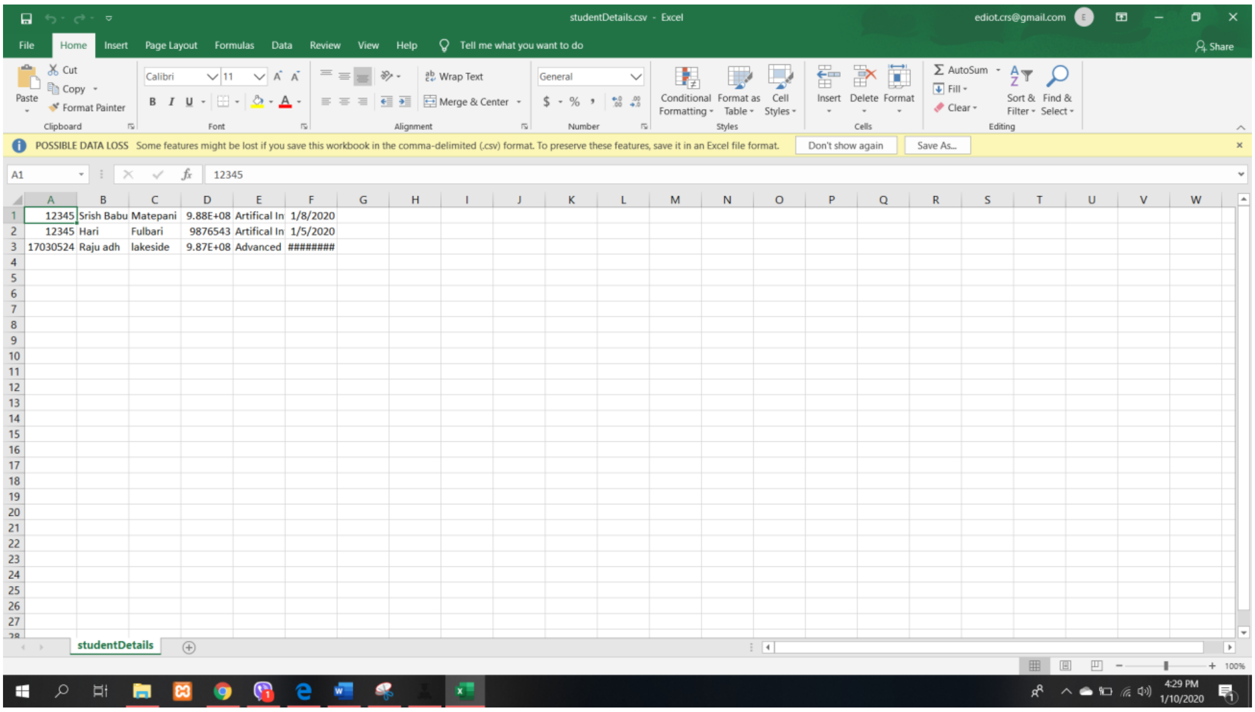
Figure 10: Student.csv file to import data in the system



ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date
12345	Hari	Fulbari	9876543	Artifical Intelligence	1/5/2020
17030524	Raju adh	lakeside	986754321	Advanced Database	2/13/2019

Figure 11: Student data after importing from csv file

After importing the csv files, it automatically saves the data of imported csv file into studentDetails.csv file.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1 12345

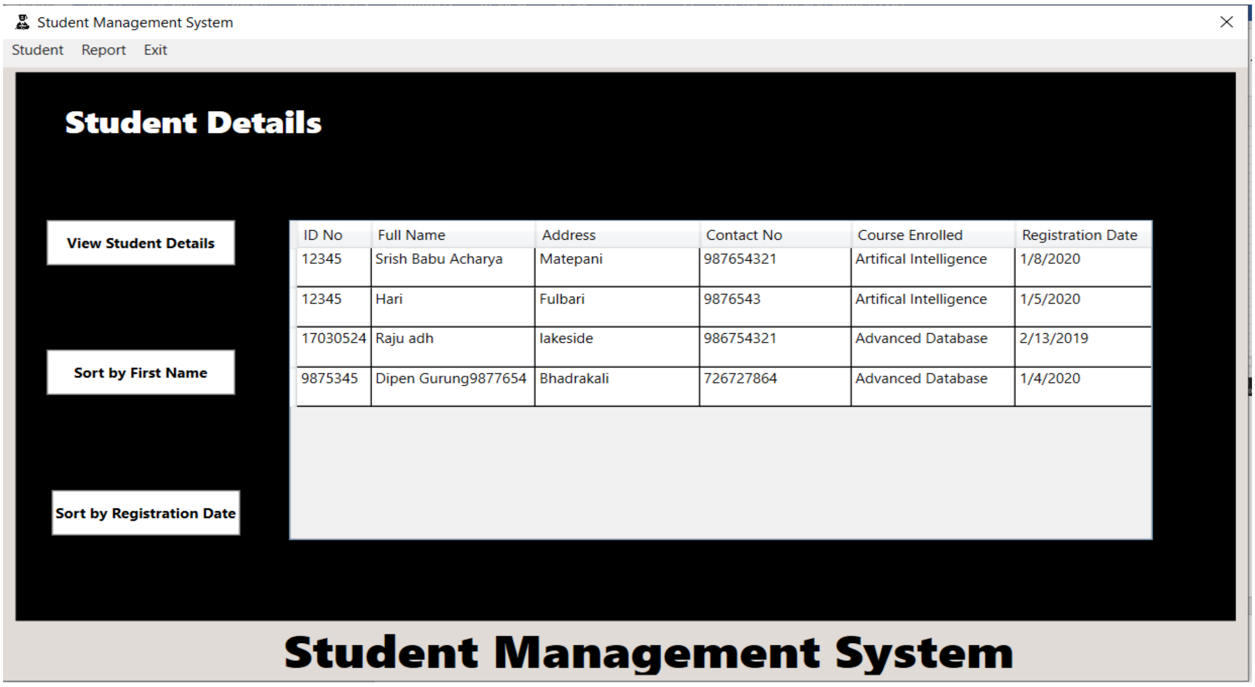
	A	B	C	D	E	F
1	12345	Srish Babu	Matepani	9.88E+08	Artifical In	1/8/2020
2	12345	Hari	Fulbari	9876543	Artifical In	1/5/2020
3	17030524	Raju adh	lakeside	9.87E+08	Advanced	########

Figure 12: Student details added after imported csv file in the system

Test Case 4:

Objective: To see student details in student details page

Result: Successfully shown student details in the datagrid


Student Management System
Student
Report
Exit
Student Details
View Student Details
Sort by First Name
Sort by Registration Date
Student Management System

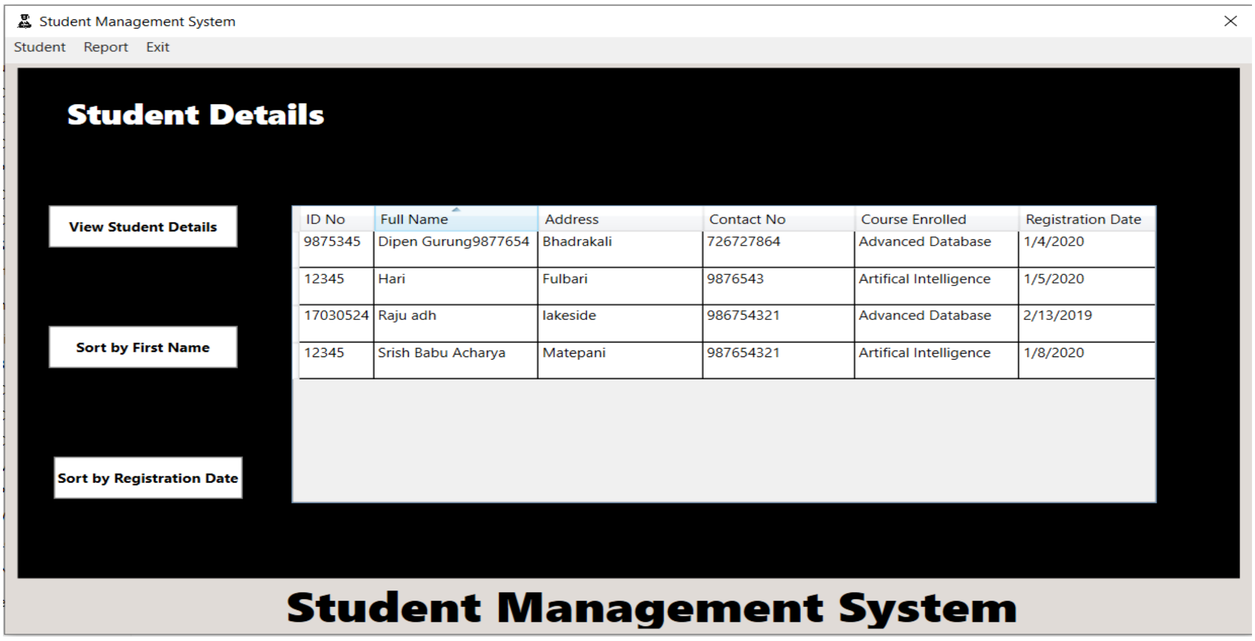
ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date
12345	Srish Babu Acharya	Matepani	987654321	Artifical Intelligence	1/8/2020
12345	Hari	Fulbari	9876543	Artifical Intelligence	1/5/2020
17030524	Raju adh	lakeside	986754321	Advanced Database	2/13/2019
9875345	Dipen Gurung9877654	Bhadrakali	726727864	Advanced Database	1/4/2020

Figure 13: Enrolled student details

Test Case 5:

Objective: To sort the student details in by their full name in alphabetical order

Result: Successfully sorted by full name


Student Management System
Student
Report
Exit
Student Details
View Student Details
Sort by First Name
Sort by Registration Date
Student Management System

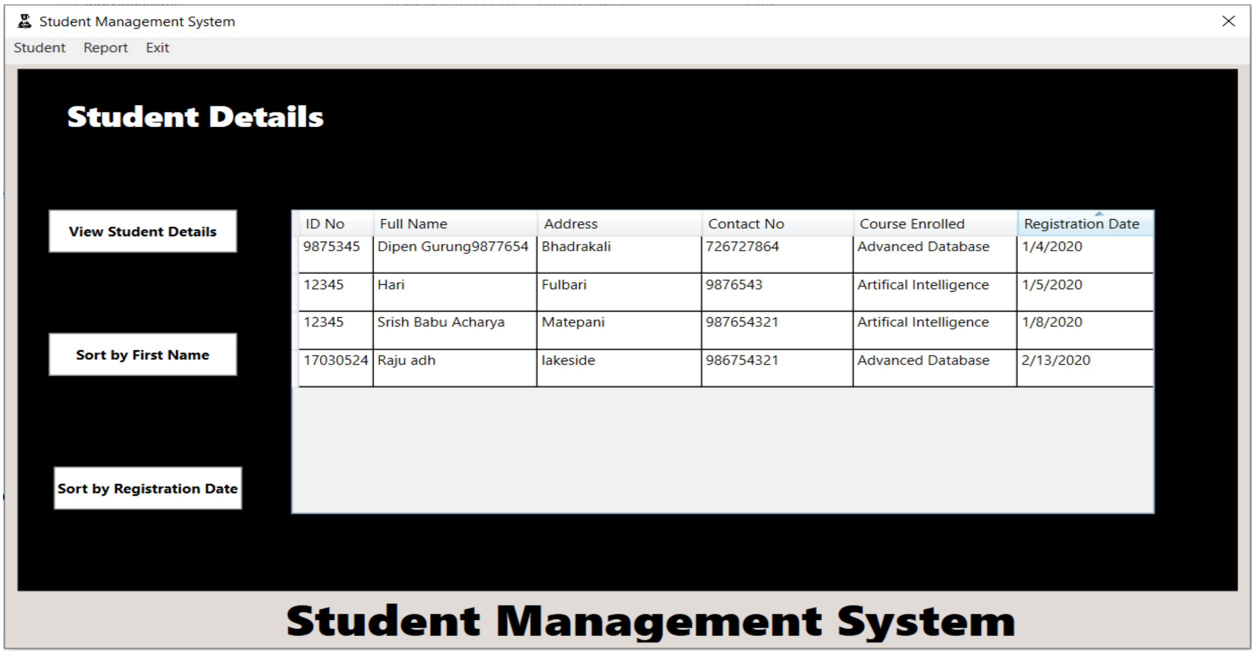
ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date
9875345	Dipen Gurung9877654	Bhadrakali	726727864	Advanced Database	1/4/2020
12345	Hari	Fulbari	9876543	Artifical Intelligence	1/5/2020
17030524	Raju adh	lakeside	986754321	Advanced Database	2/13/2019
12345	Srish Babu Acharya	Matepani	987654321	Artifical Intelligence	1/8/2020

Figure 14: Student details sorted by name

Test Case 6:

Objective: To sort the student details by registration date

Result: Successfully sorted student details by registration date.


Student Management System
Student
Report
Exit
Student Details
View Student Details
Sort by First Name
Sort by Registration Date
Student Management System

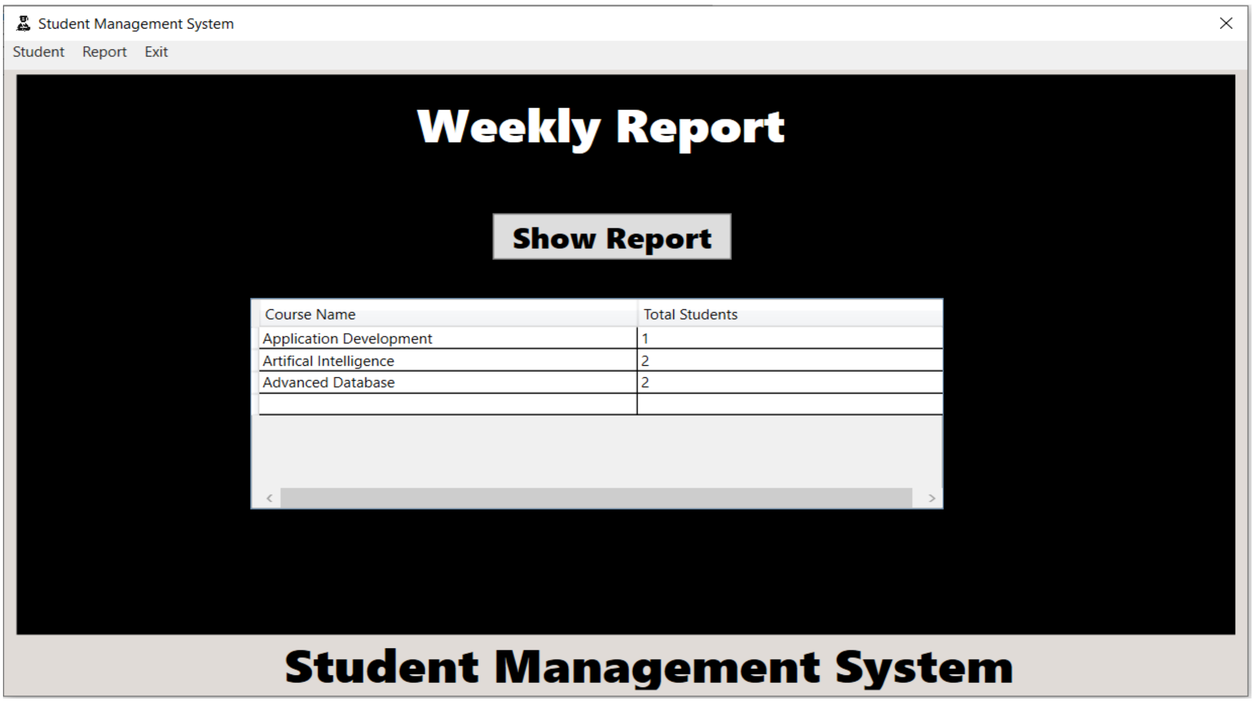
ID No	Full Name	Address	Contact No	Course Enrolled	Registration Date
9875345	Dipen Gurung9877654	Bhadrakali	726727864	Advanced Database	1/4/2020
12345	Hari	Fulbari	9876543	Artifical Intelligence	1/5/2020
12345	Srish Babu Acharya	Matepani	987654321	Artifical Intelligence	1/8/2020
17030524	Raju adh	lakeside	986754321	Advanced Database	2/13/2020

Figure 15: Sorted student details by registration date

Test Case 7:

Objective: To display weekly details

Result: successfully displayed weekly report



Course Name	Total Students
Application Development	1
Artifical Intelligence	2
Advanced Database	2

Figure 16: Weekly report of enrolled student in different courses

Test Case 8:

Objective: To generate the weekly report chart

Result: Chart successfully generated

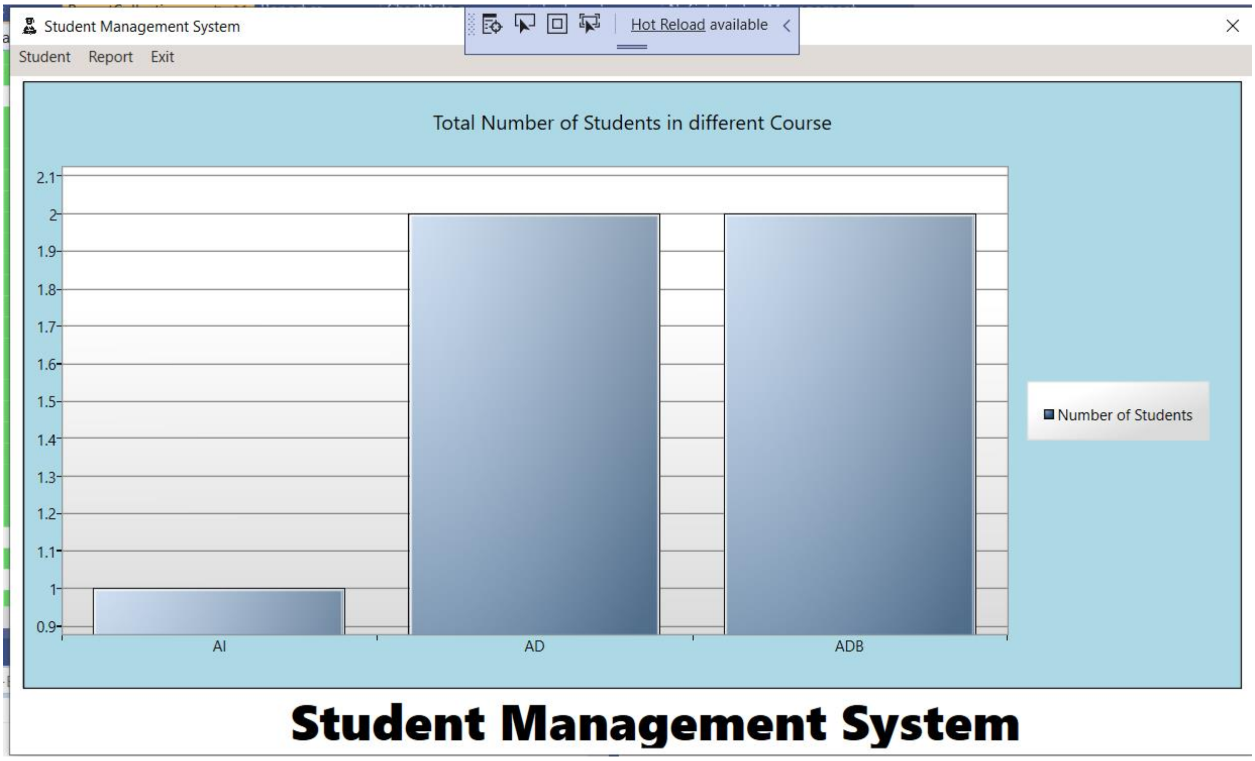


Figure 17: Bar chart showing the student counts enrolled in different courses

Algorithm and Flowchart

For Weekly Report (Algorithm)

Steps:

1. Start
2. Check whether the student enrolled data entry file exists or not.
3. If it exists, read the available data of student enrolled data entry and go to step 5.
4. If it doesn’t exist, display error message and go to step 1
5. check whether there is weekly data exists or not
6. If the data found, then retrieve the data and go to step 8
7. If f not then, display error message and go to step 1
8. Display the data as weekly report with course name and total number of student enrolled.
9. Stop.

Flowchart

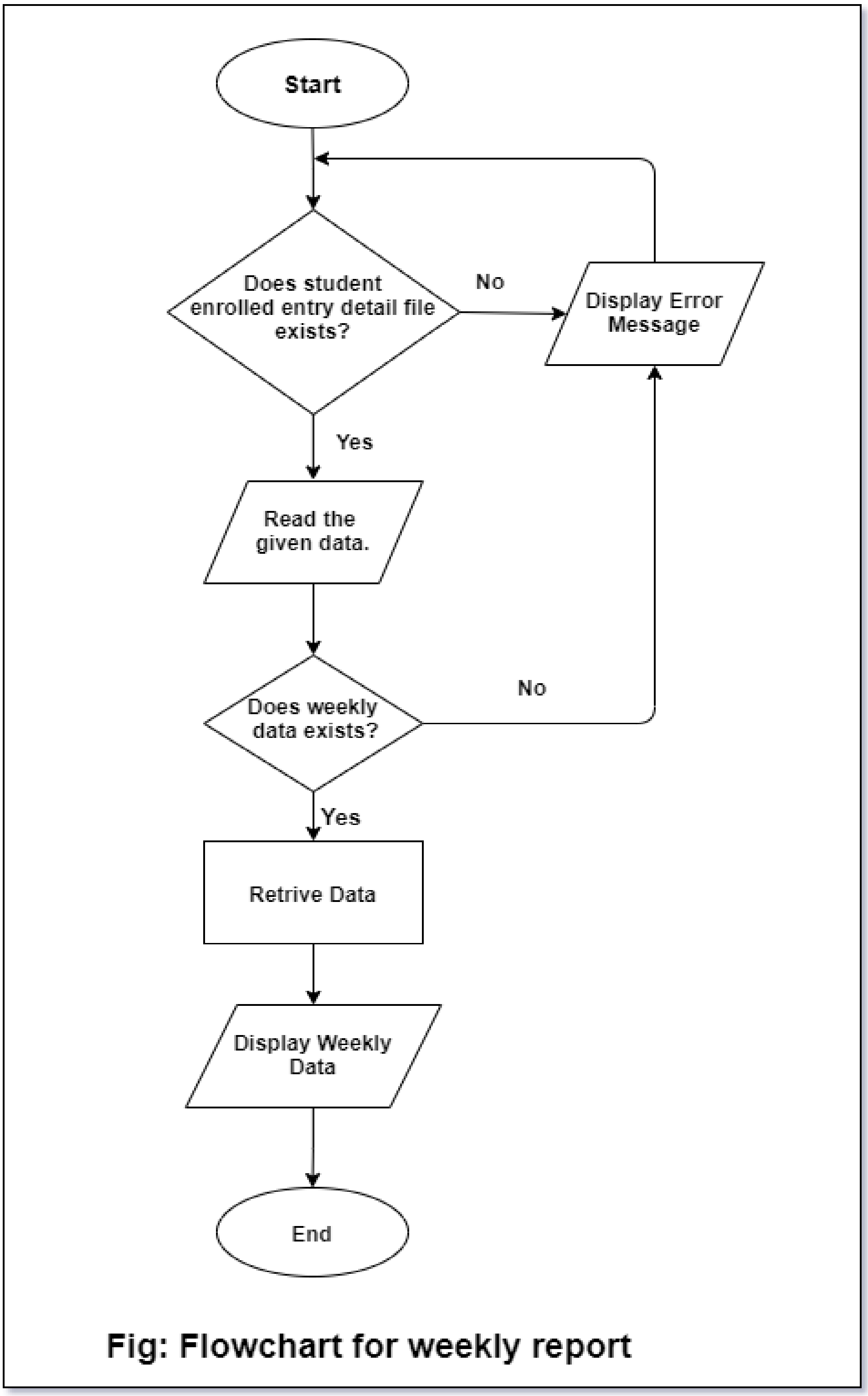


Figure 18: Flowchart of weekly report

Data Structure

List <T> Class

List<T> class represents the list of objects which can be accessed by index. It comes under the System.Collection.Generic namespace. List class can be used to create a collection of different types like integers, strings etc. List<T> class also provides the methods to search, sort, and manipulate lists.

I have used List<T> class because It is different from the arrays. A List<T> can be resized dynamically but arrays cannot. List<T> class can accept null as a valid value for reference types and it also allows duplicate elements.

Class Diagram

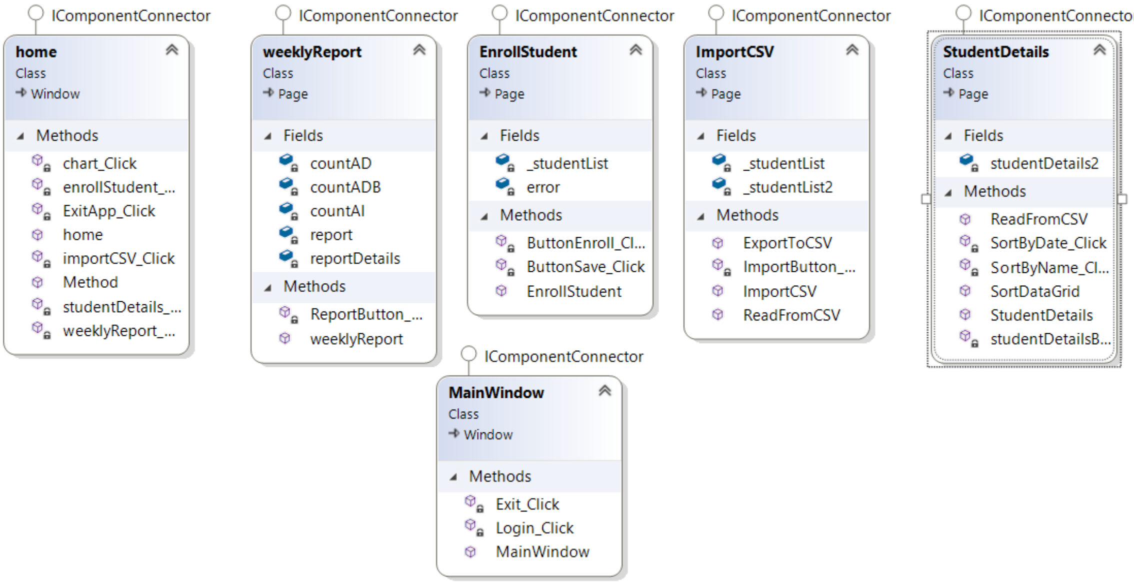


Figure 19: Class diagram of student management system

Reflection

Visual studio has been the best platform to develop student management system. Its performance is very good as it has many great features. The project has logic reflects real working environment for a schools/college. Its UI is very responsive with images, button, fonts and other features of the system. Though it is a first-time experience in the WPF-forms, I have done my best and completed the task.

The system is developed in C# programming language. Though was not easy to make a perfect user interface. I have somehow made it possible. I had drawn several mind maps while developing the program.

After the project's completion, I have gained more knowledge. I got to use my creativity and made the project possible. The time I spent working on my project was very productive.